Inizia subito

Astra Control Center

NetApp
April 25, 2024

Sommario

Inizia subito

Scopri di più su Astra Control

Astra Control è una soluzione per la gestione del ciclo di vita dei dati delle applicazioni Kubernetes che semplifica le operazioni per le applicazioni stateful. Proteggi, esegui il backup, replica e migra facilmente i carichi di lavoro Kubernetes e crea istantaneamente cloni applicativi funzionanti.

Caratteristiche

Astra Control offre funzionalità critiche per la gestione del ciclo di vita dei dati delle applicazioni Kubernetes:

- Gestire automaticamente lo storage persistente
- Creazione di snapshot e backup on-demand basati sulle applicazioni
- Automatizzare le operazioni di backup e snapshot basate su policy
- Migrare applicazioni e dati da un cluster Kubernetes a un altro
- Replica delle applicazioni su un sistema remoto utilizzando la tecnologia NetApp SnapMirror (Astra Control Center)
- Clonare le applicazioni dallo staging alla produzione
- Visualizzare lo stato di salute e protezione dell'applicazione
- Utilizzare un'interfaccia utente Web o un'API per implementare i flussi di lavoro di backup e migrazione

Modelli di implementazione

Astra Control è disponibile in due modelli di implementazione:

- **Astra Control Service**: Un servizio gestito da NetApp che offre la gestione dei dati application-aware dei cluster Kubernetes in ambienti di cloud provider multipli, oltre ai cluster Kubernetes autogestiti.
- **Astra Control Center**: Software autogestito che fornisce la gestione dei dati applicativa dei cluster Kubernetes in esecuzione nel tuo ambiente on-premise. Il centro di controllo Astra può essere installato anche in ambienti di cloud provider multipli con un backend di storage NetApp Cloud Volumes ONTAP.

	Servizio di controllo Astra	**Centro di controllo Astra**
Come viene offerto?	Come servizio cloud completamente gestito da NetApp	Come software che puoi scaricare, installare e gestire
Dove è ospitato?	Su un cloud pubblico scelto da NetApp	Sul tuo cluster Kubernetes
Come viene aggiornato?	Gestito da NetApp	Gli aggiornamenti vengono gestiti

	Servizio di controllo Astra	Centro di controllo Astra
Quali sono le distribuzioni Kubernetes supportate?	• **Cloud provider** ◦ Amazon Web Services ▪ Amazon Elastic Kubernetes Service (EKS) ◦ Google Cloud ▪ Google Kubernetes Engine (GKE) ◦ Microsoft Azure ▪ Servizio Azure Kubernetes (AKS) • **Cluster autogestiti** ◦ Kubernetes (upstream) ◦ Rancher Kubernetes Engine (RKE) ◦ Red Hat OpenShift Container Platform • **Cluster on-premise** ◦ Red Hat OpenShift Container Platform all'interno dell'hotel	• Azure Kubernetes Service su Azure Stack HCI • Google anthos • Kubernetes (upstream) • Rancher Kubernetes Engine (RKE) • Red Hat OpenShift Container Platform

	Servizio di controllo Astra	Centro di controllo Astra
Quali sono i backend di storage supportati?	• **Cloud provider** ◦ Amazon Web Services ▪ Amazon EBS ▪ Amazon FSX per NetApp ONTAP ▪ "Cloud Volumes ONTAP" ◦ Google Cloud ▪ Disco persistente di Google ▪ NetApp Cloud Volumes Service ▪ "Cloud Volumes ONTAP" ◦ Microsoft Azure ▪ Dischi gestiti Azure ▪ Azure NetApp Files ▪ "Cloud Volumes ONTAP" • **Cluster autogestiti** ◦ Amazon EBS ◦ Dischi gestiti Azure ◦ Disco persistente di Google ◦ "Cloud Volumes ONTAP" ◦ NetApp MetroCluster ◦ "Longhorn" • **Cluster on-premise** ◦ NetApp MetroCluster ◦ Sistemi NetApp ONTAP AFF e FAS ◦ NetApp ONTAP Select ◦ "Cloud Volumes ONTAP" ◦ "Longhorn"	• Sistemi NetApp ONTAP AFF e FAS • NetApp ONTAP Select • "Cloud Volumes ONTAP" • "Longhorn"

Come funziona Astra Control Service

Astra Control Service è un servizio cloud gestito da NetApp sempre attivo e aggiornato con le funzionalità più recenti. Utilizza diversi componenti per consentire la gestione del ciclo di vita dei dati delle applicazioni.

Ad alto livello, Astra Control Service funziona come segue:

- Per iniziare a utilizzare Astra Control Service, devi configurare il tuo cloud provider e registrarti per un account Astra.

    - Per i cluster GKE, Astra Control Service utilizza "NetApp Cloud Volumes Service per Google Cloud" O Google Persistent Disks come back-end di storage per i volumi persistenti.
    - Per i cluster AKS, Astra Control Service utilizza "Azure NetApp Files" O Azure Managed Disks come back-end di storage per i volumi persistenti.
    - Per i cluster Amazon EKS, Astra Control Service utilizza "Amazon Elastic Block Store" oppure "Amazon FSX per NetApp ONTAP" come back-end di storage per i volumi persistenti.
- Aggiungi il tuo primo calcolo Kubernetes ad Astra Control Service. Astra Control Service esegue le seguenti operazioni:
    - Crea un archivio di oggetti nel tuo account cloud provider, dove vengono memorizzate le copie di backup.

      In Azure, Astra Control Service crea anche un gruppo di risorse, un account di storage e chiavi per il container Blob.
    - Crea un nuovo ruolo di amministratore e un nuovo account del servizio Kubernetes sul cluster.
    - Utilizza questo nuovo ruolo di amministratore per installare link../concepts/architecture#astra-control-components[Astra Control Protivioner^] nel cluster e per creare una o più classi di storage.
    - Se utilizzi un'offerta di cloud storage NetApp come back-end dello storage, Astra Control Service utilizza Astra Control Provivisioner per il provisioning dei volumi persistenti per le tue app. Se si utilizzano dischi gestiti Amazon EBS o Azure come back-end dello storage, è necessario installare un driver CSI specifico del provider. Le istruzioni di installazione sono fornite in "Configurare Amazon Web Services" e. "Configurare Microsoft Azure con dischi gestiti Azure".
- A questo punto, è possibile aggiungere applicazioni al cluster. Il provisioning dei volumi persistenti verrà eseguito sulla nuova classe di storage predefinita.
- Quindi, utilizza Astra Control Service per gestire queste applicazioni e iniziare a creare snapshot, backup e cloni.

Il piano gratuito di Astra Control ti consente di gestire fino a 10 spazi dei nomi nel tuo account. Se desideri gestire più di 10, dovrai impostare la fatturazione eseguendo l'aggiornamento dal piano gratuito al piano Premium.

Come funziona Astra Control Center

Astra Control Center viene eseguito localmente nel tuo cloud privato.

Astra Control Center supporta i cluster Kubernetes con una classe di storage configurata da Astra Control Provisioner con un backend di storage ONTAP.

Il monitoring e la telemetria limitati (7 giorni di metriche) sono disponibili in Astra Control Center ed esportati anche in strumenti di monitoring nativi per Kubernetes (come Prometheus e Grafana) tramite end point con metriche aperte.

Il centro di controllo Astra è completamente integrato nell'ecosistema AutoSupport e Active IQ per fornire agli utenti e al supporto NetApp informazioni sulla risoluzione dei problemi e sull'utilizzo.

Puoi provare Astra Control Center utilizzando una licenza di valutazione integrata della durata di 90 giorni. Mentre stai valutando Astra Control Center, puoi ottenere supporto tramite e-mail e opzioni della community. Inoltre, puoi accedere agli articoli e alla documentazione della Knowledge base dalla dashboard di supporto all'interno del prodotto.

Per installare e utilizzare Astra Control Center, è necessario soddisfare determinati requisiti "requisiti".

Ad alto livello, Astra Control Center funziona come segue:

- Astra Control Center viene installato nel proprio ambiente locale. Scopri di più su come "Installare Astra Control Center".
- È possibile completare alcune attività di configurazione, come ad esempio:
  - Impostare la licenza.
  - Aggiungere il primo cluster.
  - Aggiungere il backend di storage rilevato quando si aggiunge il cluster.
  - Aggiungi un bucket di store di oggetti che memorizzerà i backup delle tue app.

Scopri di più su come "Configurare Astra Control Center".

È possibile aggiungere applicazioni al cluster. In alternativa, se nel cluster gestito sono già presenti alcune applicazioni, è possibile utilizzare Astra Control Center per gestirle. Quindi, utilizza Astra Control Center per creare snapshot, backup, cloni e relazioni di replica.

Per ulteriori informazioni

- "Documentazione del servizio Astra Control"
- "Documentazione di Astra Control Center"
- "Documentazione di Astra Trident"
- "Documentazione sull'API Astra Control"
- "Documentazione ONTAP"

Requisiti di Astra Control Center

Inizia verificando la preparazione del tuo ambiente operativo, dei cluster di applicazioni, delle applicazioni, delle licenze e del browser Web. Assicurati che il tuo ambiente soddisfi questi requisiti per implementare e utilizzare Astra Control Center.

Ambienti Kubernetes cluster host supportati

Astra Control Center è stato validato con i seguenti ambienti host Kubernetes:

> Assicurarsi che l'ambiente Kubernetes scelto per ospitare Astra Control Center soddisfi i requisiti di base delle risorse descritti nella documentazione ufficiale dell'ambiente.

Distribuzione di Kubernetes sul cluster host	Versioni supportate
Azure Kubernetes Service su Azure Stack HCI	Stack Azure HCI 21H2 e 22H2 con AKS 1.24.11 fino a 1.26.6
Google anthos	Da 1,15 a 1,16 (vedere Requisiti di ingresso di Google anthos)
Kubernetes (upstream)	da 1,27 a 1,29

Distribuzione di Kubernetes sul cluster host	Versioni supportate
Rancher Kubernetes Engine (RKE)	RKE 1: Versioni 1.24.17, 1.25.13, 1.26.8 con Rancher Manager 2.7.9 RKE 2: Versioni 1.23.16 e 1.24.13 con Rancher Manager 2.6.13 RKE 2: Versioni 1.24.17, 1.25.14, 1.26.9 con Rancher Manager 2.7.9
Red Hat OpenShift Container Platform	da 4,12 a 4,14

Requisiti delle risorse del cluster host

Astra Control Center richiede le seguenti risorse oltre ai requisiti delle risorse dell'ambiente:

> Questi requisiti presuppongono che Astra Control Center sia l'unica applicazione in esecuzione nell'ambiente operativo. Se nell'ambiente sono in esecuzione applicazioni aggiuntive, modificare di conseguenza questi requisiti minimi.

- **CPU Extensions**: Le CPU di tutti i nodi dell'ambiente di hosting devono avere le estensioni AVX abilitate.
- **Nodi di lavoro**: Almeno 3 nodi di lavoro in totale, con 4 core CPU e 12 GB di RAM ciascuno
- **Requisiti del cluster VMware Tanzu Kubernetes Grid**: Quando si ospita Astra Control Center su un cluster VMware Tanzu Kubernetes Grid (TKG) o Tanzu Kubernetes Grid Integrated Edition (TKGi), tenere presente le seguenti considerazioni.
  - Il token del file di configurazione predefinito di VMware TKG e TKGi scade dieci ore dopo l'implementazione. Se si utilizzano prodotti del portfolio Tanzu, è necessario generare un file di configurazione del cluster Tanzu Kubernetes con un token non in scadenza per evitare problemi di connessione tra Astra Control Center e cluster di applicazioni gestiti. Per istruzioni, visitare il sito "Documentazione del prodotto VMware NSX-T Data Center."
  - Utilizzare `kubectl get nsxlbmonitors -A` per verificare se è già stato configurato un monitor dei servizi per accettare il traffico in entrata. Se ne esiste uno, non installare MetalLB, perché il monitor di servizio esistente sovrascriverà qualsiasi nuova configurazione del bilanciamento del carico.
  - Disattivare l'applicazione della classe di storage predefinita TKG o TKGi su qualsiasi cluster di applicazioni che deve essere gestito da Astra Control. Per eseguire questa operazione, modificare il `TanzuKubernetesCluster` risorsa sul cluster dello spazio dei nomi.
  - Quando implementi Astra Control Center in un ambiente TKG o TKGi, tieni presente i requisiti specifici di Astra Control Provisioner:
    - Il cluster deve supportare workload con privilegi.
    - Il `--kubelet-dir` flag deve essere impostato sulla posizione della directory di kubelet. Per impostazione predefinita, questo è `/var/vcap/data/kubelet`.
    - Specificare la posizione del kubelet utilizzando `--kubelet-dir` È noto per lavorare con Trident Operator, Helm e. `tridentctl` implementazioni.

Requisiti mesh di servizio

Si consiglia vivamente di installare una versione vanilla supportata della mesh del servizio Istio sul cluster host Astra Control Center. Fare riferimento a. "versioni supportate" Per le versioni supportate di Istio. Le versioni con marchio di Istio Service Mesh, come OpenShift Service Mesh, non sono validate con Astra Control Center.

Per integrare Astra Control Center con la mesh di servizio Istio installata sul cluster host, è necessario eseguire l'integrazione come parte di Astra Control Center "installazione" e non indipendente da questo processo.

> L'installazione e l'utilizzo di Astra Control Center senza la configurazione di una mesh di servizio nel cluster host ha potenzialmente serie implicazioni per la sicurezza.

Astra Trident

Se intendi utilizzare Astra Trident al posto di Astra Control Provivisioner con questa release, sono supportate Astra Trident 23,04 e versioni successive. Astra Control Center richiede Astra Control provisioner nelle versioni future.

Astra Control provisioner

Per utilizzare le funzionalità di storage avanzate di Astra Control Provivisioner, devi installare Astra Trident 23,10 o versioni successive e abilitare "Funzionalità Astra Control Provisioner". Per utilizzare la funzionalità più recente di Astra Control Provisioner, avrai bisogno delle versioni più recenti di Astra Trident e Astra Control Center.

- **Versione minima di Astra Control Provisioner da utilizzare con Astra Control Center**: Astra Control Provisioner 23,10 o versione successiva installato e configurato.

Configurazione di ONTAP con Astra Trident

- **Storage class**: Configurare almeno una classe di archiviazione nel cluster. Se viene configurata una classe di storage predefinita, assicurarsi che sia l'unica classe di storage con la designazione predefinita.
- **Driver di storage e nodi di lavoro**: Assicurarsi di configurare i nodi di lavoro nel cluster con i driver di storage appropriati in modo che i pod possano interagire con lo storage backend. Centro di controllo Astra supporta i seguenti driver ONTAP forniti da Astra Trident:
  - `ontap-nas`
  - `ontap-san`
  - `ontap-san-economy` (la replica dell'applicazione non è disponibile con questo tipo di classe di archiviazione)
  - `ontap-nas-economy` (le snapshot e le policy di replica delle applicazioni non sono disponibili con questo tipo di classe di storage)

Back-end dello storage

Assicurarsi di disporre di un backend supportato con capacità sufficiente.

- **Capacità di back-end dello storage richiesta**: Almeno 500 GB disponibili
- **Backend supportati**: Astra Control Center supporta i seguenti backend di storage:
  - NetApp ONTAP 9.9.1 o sistemi AFF, FAS e ASA successivi
  - NetApp ONTAP Select 9.9.1 o versione successiva
  - NetApp Cloud Volumes ONTAP 9.9.1 o versione successiva
  - (Per l'anteprima tecnica Astra Control Center) NetApp ONTAP 9.10.1 o versioni successive per le operazioni di protezione dei dati fornite come anteprima tecnica

- Longhorn 1.5.0 o successivo
  - Richiede la creazione manuale di un oggetto VolumeSnapshotClass. Fare riferimento a. "Documentazione di Longhorn" per istruzioni.
- NetApp MetroCluster
  - I cluster Kubernetes gestiti devono essere in una configurazione stretch.
- Backend di storage disponibili con cloud provider supportati

Licenze ONTAP

Per utilizzare il centro di controllo Astra, verificare di disporre delle seguenti licenze ONTAP, a seconda delle operazioni da eseguire:

- FlexClone
- SnapMirror: Opzionale. Necessario solo per la replica su sistemi remoti utilizzando la tecnologia SnapMirror. Fare riferimento a. "Informazioni sulla licenza SnapMirror".
- Licenza S3: Opzionale. Necessario solo per i bucket ONTAP S3

Per verificare se il sistema ONTAP dispone delle licenze richieste, fare riferimento a. "Gestire le licenze ONTAP".

NetApp MetroCluster

Quando utilizzi NetApp MetroCluster come back-end dello storage, devi quanto segue:

- Specifica una LIF di gestione SVM come opzione di backend nel driver Astra Trident che utilizzi
- Assicurarsi di disporre della licenza ONTAP appropriata

Per configurare la LIF MetroCluster, fai riferimento a queste opzioni ed esempi per ogni driver:

- "SAN"
- "NAS"

Licenza Astra Control Center

Astra Control Center richiede una licenza Astra Control Center. Quando si installa Astra Control Center, viene già attivata una licenza di valutazione integrata di 90 giorni per 4,800 unità CPU. Se hai bisogno di una maggiore capacità o di termini di valutazione diversi, o se desideri passare a una licenza completa, puoi ottenere una licenza di valutazione o una licenza completa diversa da NetApp. Hai bisogno di una licenza per proteggere le tue applicazioni e i tuoi dati.

Puoi provare Astra Control Center registrandoti per una prova gratuita. Puoi iscriverti registrandoti "qui".

Per impostare la licenza, fare riferimento a. "utilizzare una licenza di valutazione di 90 giorni".

Per ulteriori informazioni sul funzionamento delle licenze, fare riferimento a. "Licensing".

Requisiti di rete

Configura il tuo ambiente operativo per garantire che Astra Control Center possa comunicare correttamente. Sono necessarie le seguenti configurazioni di rete:

- **Indirizzo FQDN**: È necessario disporre di un indirizzo FQDN per Astra Control Center.
- **Accesso a Internet**: È necessario determinare se si dispone di accesso esterno a Internet. In caso contrario, alcune funzionalità potrebbero essere limitate, ad esempio l'invio di pacchetti di supporto al "Sito di supporto NetApp".
- **Port Access**: L'ambiente operativo che ospita Astra Control Center comunica utilizzando le seguenti porte TCP. Assicurarsi che queste porte siano consentite attraverso qualsiasi firewall e configurare i firewall in modo da consentire qualsiasi traffico HTTPS in uscita dalla rete Astra. Alcune porte richiedono la connettività tra l'ambiente che ospita Astra Control Center e ciascun cluster gestito (annotato dove applicabile).

> Puoi implementare Astra Control Center in un cluster Kubernetes dual-stack, mentre Astra Control Center può gestire le applicazioni e i back-end di storage configurati per il funzionamento dual-stack. Per ulteriori informazioni sui requisiti del cluster dual-stack, vedere "Documentazione Kubernetes".

Origine	Destinazione	Porta	Protocollo	Scopo
PC client	Centro di controllo Astra	443	HTTPS	Accesso UI/API - assicurarsi che questa porta sia aperta in entrambe le direzioni tra Astra Control Center e il sistema utilizzato per accedere ad Astra Control Center
Metriche consumer	Nodo di lavoro Astra Control Center	9090	HTTPS	Comunicazione dei dati delle metriche - garantire che ciascun cluster gestito possa accedere a questa porta sul cluster che ospita Astra Control Center (è richiesta una comunicazione bidirezionale)
Centro di controllo Astra	Provider di bucket di storage Amazon S3	443	HTTPS	Comunicazione con lo storage Amazon S3
Centro di controllo Astra	NetApp AutoSupport (https://support.netapp.com)	443	HTTPS	Comunicazioni NetApp AutoSupport

Origine	Destinazione	Porta	Protocollo	Scopo
Centro di controllo Astra	Cluster Kubernetes gestito	443/6443 **NOTA**: La porta utilizzata dal cluster gestito può variare a seconda del cluster. Fare riferimento alla documentazione fornita dal fornitore del software per cluster.	HTTPS	Comunicazione con il cluster gestito - assicurarsi che questa porta sia aperta in entrambi i modi tra il cluster che ospita Astra Control Center e ciascun cluster gestito

Ingresso per cluster Kubernetes on-premise

È possibile scegliere il tipo di ingresso di rete utilizzato da Astra Control Center. Per impostazione predefinita, Astra Control Center implementa il gateway Astra Control Center (servizio/traefik) come risorsa a livello di cluster. Astra Control Center supporta anche l'utilizzo di un servizio di bilanciamento del carico, se consentito nel tuo ambiente. Se si preferisce utilizzare un servizio di bilanciamento del carico e non ne si dispone già di uno configurato, è possibile utilizzare il bilanciamento del carico MetalLB per assegnare automaticamente un indirizzo IP esterno al servizio. Nella configurazione del server DNS interno, puntare il nome DNS scelto per Astra Control Center sull'indirizzo IP con bilanciamento del carico.

> Il bilanciamento del carico deve utilizzare un indirizzo IP situato nella stessa subnet degli indirizzi IP del nodo di lavoro di Astra Control Center.

Per ulteriori informazioni, fare riferimento a. "Impostare l'ingresso per il bilanciamento del carico".

Requisiti di ingresso di Google anthos

Quando si ospita Astra Control Center su un cluster Google anthos, Google anthos include il bilanciamento del carico MetalLB e il servizio di ingresso Istio per impostazione predefinita, consentendo di utilizzare semplicemente le funzionalità di ingresso generiche di Astra Control Center durante l'installazione. Fare riferimento a. "Documentazione di installazione di Astra Control Center" per ulteriori informazioni.

Browser Web supportati

Astra Control Center supporta versioni recenti di Firefox, Safari e Chrome con una risoluzione minima di 1280 x 720.

Requisiti aggiuntivi per i cluster di applicazioni

Se si prevede di utilizzare queste funzionalità di Astra Control Center, tenere presenti questi requisiti:

- **Requisiti del cluster applicativo**: "Requisiti di gestione del cluster"
  - **Requisiti delle applicazioni gestite**: "Requisiti di gestione delle applicazioni"
  - **Requisiti aggiuntivi per la replica delle applicazioni**: "Prerequisiti per la replica"

Cosa succederà

Visualizzare il "avvio rapido" panoramica.

Avvio rapido per Astra Control Center

Ecco una panoramica dei passaggi necessari per iniziare a utilizzare Astra Control Center. I collegamenti all'interno di ogni passaggio consentono di accedere a una pagina che fornisce ulteriori dettagli.

1 Esaminare i requisiti del cluster Kubernetes

Assicurarsi che l'ambiente soddisfi i seguenti requisiti:

Cluster Kubernetes

- "Assicurarsi che il cluster host soddisfi i requisiti dell'ambiente operativo"
- "Configurare l'ingresso per il bilanciamento del carico dei cluster Kubernetes on-premise"

Integrazione dello storage

- "Assicurati che il tuo ambiente includa Astra Control Provisioner"
- "Abilita le funzionalità avanzate di gestione e provisioning dello storage di Astra Control Provisioner"
- "Preparare i nodi di lavoro del cluster"
- "Configurare i backend di storage"
- "Configurare le classi di archiviazione"
- "Installare un controller per lo snapshot del volume"
- "Creare una classe di snapshot di volume"

Credenziali ONTAP

- "Configurare le credenziali ONTAP"

2 Scaricare e installare Astra Control Center

Completare le seguenti attività di installazione:

- "Scarica Astra Control Center dalla pagina di download del sito di supporto NetApp"
- Ottenere il file di licenza NetApp:
  - Se si sta valutando Astra Control Center, è già inclusa una licenza di valutazione integrata
  - "Se si è già acquistato Astra Control Center, generare il file di licenza"
- "Installare Astra Control Center"
- "Eseguire ulteriori procedure di configurazione opzionali"

3 Completare alcune attività di configurazione iniziali

Completare alcune attività di base per iniziare:

- "Aggiungere una licenza"

- "Prepara il tuo ambiente per la gestione dei cluster"
- "Aggiungere un cluster"
- "Aggiungere un backend di storage"
- "Aggiungi un bucket"

4 Utilizzare Astra Control Center

Una volta completata la configurazione di Astra Control Center, utilizzare l'interfaccia utente di Astra Control o il "API di controllo Astra" per iniziare a gestire e proteggere le applicazioni:

- "Gestire gli account": Utenti, ruoli, LDAP, credenziali e altro ancora.
- "Gestire le notifiche"
- "Gestire le applicazioni": Definire le risorse da gestire.
- "Proteggi le app": Configurare le policy di protezione e replicare, clonare e migrare le applicazioni.

Per ulteriori informazioni

- "Utilizzare l'API di controllo Astra"
- "Aggiornare Astra Control Center"
- "Ottieni assistenza con Astra Control"

Panoramica dell'installazione

Scegliere e completare una delle seguenti procedure di installazione di Astra Control Center:

- "Installare Astra Control Center utilizzando il processo standard"
- "(Se utilizzi Red Hat OpenShift) Installa Astra Control Center usando OpenShift OperatorHub"
- "Installare il centro di controllo Astra con un backend di storage Cloud Volumes ONTAP"

A seconda dell'ambiente in uso, potrebbe essere necessaria una configurazione aggiuntiva dopo l'installazione di Astra Control Center:

- "Configurare Astra Control Center dopo l'installazione"

Installare Astra Control Center utilizzando il processo standard

Per installare Astra Control Center, scaricare le immagini di installazione ed eseguire i seguenti passaggi. È possibile utilizzare questa procedura per installare Astra Control Center in ambienti connessi a Internet o con connessione ad aria.

Per una dimostrazione del processo di installazione di Astra Control Center, vedere "questo video".

Prima di iniziare

- **Soddisfare i requisiti ambientali**: "Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center".

> Implementare Astra Control Center in un terzo dominio di errore o in un sito secondario. Questa opzione è consigliata per la replica delle applicazioni e il disaster recovery perfetto.

- **Garantire servizi integri**: Controllare che tutti i servizi API siano in buono stato e disponibili:

```
kubectl get apiservices
```

- **Assicurarsi che un FQDN instradabile**: Il FQDN Astra che si intende utilizzare può essere instradato al cluster. Ciò significa che si dispone di una voce DNS nel server DNS interno o si sta utilizzando un percorso URL principale già registrato.
- **Configure cert manager**: Se nel cluster esiste già un cert manager, è necessario eseguirne alcuni "fasi preliminari" In modo che Astra Control Center non tenti di installare il proprio cert manager. Per impostazione predefinita, Astra Control Center installa il proprio cert manager durante l'installazione.
- * (Solo driver SAN ONTAP) Abilita multipath*: Se stai utilizzando un driver SAN ONTAP, assicurati che multipath sia abilitato su tutti i tuoi cluster Kubernetes.

È inoltre necessario considerare quanto segue:

- **Ottenere l'accesso al Registro di sistema dell'immagine di controllo Astra di NetApp**:

  È possibile ottenere le immagini di installazione e i miglioramenti delle funzionalità per Astra Control, come Astra Control provisioner, dal registro delle immagini di NetApp.

  a. Registrare l'ID dell'account Astra Control necessario per accedere al Registro di sistema.

     Puoi visualizzare l'ID dell'account nell'interfaccia utente Web di Astra Control Service. Selezionare l'icona a forma di figura in alto a destra nella pagina, selezionare **accesso API** e annotare l'ID account.

  b. Nella stessa pagina, selezionare **generate API token**, copiare la stringa del token API negli Appunti e salvarla nell'editor.

  c. Accedere al registro Astra Control:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

- **Installare una mesh di servizio per comunicazioni sicure**: Si consiglia vivamente di proteggere i canali di comunicazione del cluster host Astra Control utilizzando un "mesh di servizio supportata".

> L'integrazione di Astra Control Center con una mesh di servizio può essere eseguita solo durante Astra Control Center "installazione" e non indipendente da questo processo. Il passaggio da un ambiente con mesh a un ambiente senza mesh non è supportato.

  Per l'uso della mesh del servizio Istio, è necessario effettuare le seguenti operazioni:

  - Aggiungere un `istio-injection:enabled` etichetta Al namespace Astra prima di implementare Astra Control Center.
  - Utilizzare `Generic` impostazione ingresso e fornire un ingresso alternativo per bilanciamento del carico esterno.
  - Per i cluster Red Hat OpenShift, è necessario definire `NetworkAttachmentDefinition` Su tutti i

namespace Astra Control Center associati (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` per i cluster di applicazioni o qualsiasi namespace personalizzato che sia stato sostituito).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Fasi

Per installare Astra Control Center, procedere come segue:

- Scarica ed estrai Astra Control Center
- Completare ulteriori passaggi se si utilizza un registro locale
- Impostare namespace e secret per i registri con requisiti di autenticazione
- Installare l'operatore del centro di controllo Astra
- Configurare Astra Control Center
- Completare l'installazione dell'Astra Control Center e dell'operatore
- Verificare lo stato del sistema
- Impostare l'ingresso per il bilanciamento del carico
- Accedere all'interfaccia utente di Astra Control Center

> Non eliminare l'operatore di Astra Control Center (ad esempio, `kubectl delete -f astra_control_center_operator_deploy.yaml`) In qualsiasi momento durante l'installazione o il funzionamento di Astra Control Center per evitare di eliminare i pod.

Scarica ed estrai Astra Control Center

Scarica le immagini di Astra Control Center da una delle seguenti posizioni:

- **Registro di sistema dell'immagine del servizio di controllo Astra**: Utilizzare questa opzione se non si utilizza un registro locale con le immagini del centro di controllo Astra o se si preferisce questo metodo per il download del pacchetto dal sito di supporto NetApp.
- **Sito di supporto NetApp**: Utilizzare questa opzione se si utilizza un registro locale con le immagini del Centro di controllo Astra.

Registro delle immagini di Astra Control

1. Effettua l'accesso ad Astra Control Service.
2. Nella Dashboard, selezionare **distribuire un'istanza autogestita di Astra Control**.
3. Seguire le istruzioni per accedere al registro delle immagini di Astra Control, estrarre l'immagine di installazione di Astra Control Center ed estrarre l'immagine.

Sito di supporto NetApp

1. Scarica il bundle contenente Astra Control Center (`astra-control-center-[version].tar.gz`) da "Pagina di download di Astra Control Center".
2. (Consigliato ma opzionale) Scarica il bundle di certificati e firme per Astra Control Center (`astra-control-center-certs-[version].tar.gz`) per verificare la firma del bundle.

   ```
   tar -vxzf astra-control-center-certs-[version].tar.gz
   ```

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig astra-
   control-center-[version].tar.gz
   ```

   Viene visualizzato l'output `Verified OK` una volta completata la verifica.

3. Estrarre le immagini dal bundle Astra Control Center:

   ```
   tar -vxzf astra-control-center-[version].tar.gz
   ```

Completare ulteriori passaggi se si utilizza un registro locale

Se si intende inviare il pacchetto Astra Control Center al registro locale, è necessario utilizzare il plugin della riga di comando di NetApp Astra kubectl.

Installare il plug-in NetApp Astra kubectl

Completare questi passaggi per installare il più recente plugin della riga di comando di NetApp Astra kubectl.

Prima di iniziare

NetApp fornisce binari per plug-in per diverse architetture CPU e sistemi operativi. Prima di eseguire questa attività, è necessario conoscere la CPU e il sistema operativo in uso.

Se il plug-in è già stato installato da un'installazione precedente, "assicurarsi di disporre della versione più recente" prima di completare questa procedura.

Fasi

1. Elencare i binari disponibili per il plugin NetApp Astra kubectl:

   > La libreria di plugin kubectl fa parte del bundle tar e viene estratta nella cartella `kubectl-astra`.

   ```
   ls kubectl-astra/
   ```

2. Spostare il file necessario per il sistema operativo e l'architettura della CPU nel percorso corrente e rinominarlo `kubectl-astra`:

   ```
   cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
   ```

Aggiungere le immagini al registro

1. Se si prevede di inviare il pacchetto Astra Control Center al registro locale, completare la sequenza di passaggi appropriata per il motore del contenitore:

Docker

a. Passare alla directory root del tarball. Viene visualizzata la `acc.manifest.bundle.yaml` file e queste directory:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Trasferire le immagini del pacchetto nella directory delle immagini di Astra Control Center nel registro locale. Eseguire le seguenti sostituzioni prima di eseguire `push-images` comando:

- Sostituire <BUNDLE_FILE> con il nome del file bundle di controllo Astra (`acc.manifest.bundle.yaml`).
- Sostituire <MY_FULL_REGISTRY_PATH> con l’URL del repository Docker; ad esempio, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
- Sostituire <MY_REGISTRY_USER> con il nome utente.
- Sostituire <MY_REGISTRY_TOKEN> con un token autorizzato per il registro.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

a. Passare alla directory root del tarball. Vengono visualizzati il file e la directory seguenti:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Accedere al Registro di sistema:

```
podman login <YOUR_REGISTRY>
```

c. Preparare ed eseguire uno dei seguenti script personalizzato per la versione di Podman utilizzata. Sostituire <MY_FULL_REGISTRY_PATH> con l'URL del repository che include le sottodirectory.

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

Il percorso dell'immagine creato dallo script deve essere simile al seguente, a seconda della configurazione del Registro di sistema:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Modificare la directory:

```
cd manifests
```

Impostare namespace e secret per i registri con requisiti di autenticazione

1. Esportare il file kubeconfig per il cluster host Astra Control Center:

```
export KUBECONFIG=[file path]
```

> Prima di completare l'installazione, assicurarsi che kubeconfig punti al cluster in cui si desidera installare Astra Control Center.

2. Se si utilizza un registro che richiede l'autenticazione, è necessario effettuare le seguenti operazioni:

   a. Creare il `netapp-acc-operator` spazio dei nomi:

```
kubectl create ns netapp-acc-operator
```

   b. Creare un segreto per `netapp-acc-operator` namespace. Aggiungere informazioni su Docker ed eseguire il seguente comando:

> Il segnaposto `your_registry_path` deve corrispondere alla posizione delle immagini caricate in precedenza (ad esempio, `[Registry_URL]/netapp/astra/astracc/24.02.0-69`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

+

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=[your_registry_path] --docker
-username=[username] --docker-password=[token]
```

+

> Se si elimina lo spazio dei nomi dopo la generazione del segreto, ricreare lo spazio dei nomi e rigenerare il segreto per lo spazio dei nomi.

   a. Creare il `netapp-acc` namespace (o personalizzato).

```
kubectl create ns [netapp-acc or custom namespace]
```

   b. Creare un segreto per `netapp-acc` namespace (o personalizzato). Aggiungere informazioni su Docker ed eseguire uno dei comandi appropriati in base alle preferenze del Registro di sistema:

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-
acc or custom namespace] --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-
acc or custom namespace] --docker-server=[your_registry_path]
--docker-username=[username] --docker-password=[token]
```

Installare l'operatore del centro di controllo Astra

1. (Solo registri locali) se si utilizza un registro locale, completare i seguenti passaggi:

   a. Aprire il programma YAML di distribuzione dell'operatore Astra Control Center:

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   Un YAML di esempio annotato segue questi passaggi.

   b. Se si utilizza un registro che richiede l'autenticazione, sostituire la riga predefinita di `imagePullSecrets: []` con i seguenti elementi:

   ```
   imagePullSecrets: [{name: astra-registry-cred}]
   ```

   c. Cambiare `ASTRA_IMAGE_REGISTRY` per `kube-rbac-proxy` al percorso del registro in cui sono state inviate le immagini in a. passaggio precedente.

   d. Cambiare `ASTRA_IMAGE_REGISTRY` per `acc-operator-controller-manager` al percorso del registro in cui sono state inviate le immagini in a. passaggio precedente.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
```

```
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
```

```
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

2. Installare l'operatore del centro di controllo Astra:

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

Espandi per la risposta di esempio:

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.as
tra.netapp.io created
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
created
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role
created
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding created
configmap/acc-operator-manager-config created
service/acc-operator-controller-manager-metrics-service created
deployment.apps/acc-operator-controller-manager created
```

3. Verificare che i pod siano in esecuzione:

```
kubectl get pods -n netapp-acc-operator
```

Configurare Astra Control Center

1. Modificare il file delle risorse personalizzate (CR) di Astra Control Center

(`astra_control_center.yaml`) per creare account, supporto, registro e altre configurazioni necessarie:

```
vim astra_control_center.yaml
```

Un YAML di esempio annotato segue questi passaggi.

2. Modificare o confermare le seguenti impostazioni:

Nome account

Impostazione	Guida	Tipo	Esempio
`accountName`	Modificare il `accountName` Stringa al nome che si desidera associare all'account Astra Control Center. Può essere presente un solo nome account.	stringa	`Example`

AstraVersion

Impostazione	Guida	Tipo	Esempio
`astraVersion`	La versione di Astra Control Center da implementare. Non è necessaria alcuna azione per questa impostazione, in quanto il valore verrà pre-compilato.	stringa	`24.02.0-69`

AstraAddress

Impostazione	Guida	Tipo	Esempio
`astraAddress`	Modificare il `astraAddress` Inserire l'FQDN (consigliato) o l'indirizzo IP che si desidera utilizzare nel browser per accedere ad Astra Control Center. Questo indirizzo definisce il modo in cui Astra Control Center verrà trovato nel data center e corrisponde allo stesso FQDN o indirizzo IP fornito dal bilanciamento del carico al termine dell'operazione "Requisiti di Astra Control Center". NOTA: Non utilizzare `http://` oppure `https://` nell'indirizzo. Copiare questo FQDN per utilizzarlo in un passo successivo.	stringa	`astra.example.com`

AutoSupport

Le selezioni effettuate in questa sezione determinano se parteciperai all'applicazione di supporto proattivo di NetApp, NetApp Active IQ, e dove vengono inviati i dati. È necessaria una connessione a Internet (porta 442) e tutti i dati di supporto sono resi anonimi.

Impostazione	Utilizzare	Guida	Tipo	Esempio
`autoSupport.enrolled`	Entrambi `enrolled` oppure `url` i campi devono essere selezionati	Cambiare `enrolled` Per AutoSupport a. `false` per i siti senza connettività internet o senza retain `true` per i siti connessi. Un'impostazione di `true` Consente l'invio di dati anonimi a NetApp a scopo di supporto. L'elezione predefinita è `false` E indica che non verranno inviati dati di supporto a NetApp.	Booleano	`false` (valore predefinito)
`autoSupport.url`	Entrambi `enrolled` oppure `url` i campi devono essere selezionati	Questo URL determina dove verranno inviati i dati anonimi.	stringa	`https://support.netapp.com/asupprod/post/1.0/postAsup`

e-mail

Impostazione	Guida	Tipo	Esempio
`email`	Modificare il `email` stringa all'indirizzo iniziale predefinito dell'amministratore. Copiare questo indirizzo e-mail per utilizzarlo in passo successivo. Questo indirizzo e-mail verrà utilizzato come nome utente per l'account iniziale per accedere all'interfaccia utente e verrà notificato degli eventi in Astra Control.	stringa	`admin@example.com`

Nome

Impostazione	Guida	Tipo	Esempio
`firstName`	Il nome dell'amministratore iniziale predefinito associato all'account Astra. Il nome utilizzato qui sarà visibile in un'intestazione dell'interfaccia utente dopo il primo accesso.	stringa	`SRE`

Cognome

Impostazione	Guida	Tipo	Esempio
`lastName`	Il cognome dell'amministratore iniziale predefinito associato all'account Astra. Il nome utilizzato qui sarà visibile in un'intestazione dell'interfaccia utente dopo il primo accesso.	stringa	`Admin`

ImageRegistry

Le selezioni effettuate in questa sezione definiscono il registro delle immagini container che ospita le immagini dell'applicazione Astra, Astra Control Center Operator e il repository Astra Control Center Helm.

Impostazione	Utilizzare	Guida	Tipo	Esempio
`imageRegistry.name`	Obbligatorio	Nome del registro delle immagini di Astra Control che ospita tutte le immagini richieste per distribuire Astra Control Center. Il valore viene precompilato e non è richiesta alcuna azione, a meno che non sia stato configurato un registro locale. Per un registro locale, sostituire questo valore esistente con il nome del registro delle immagini in cui sono state inserite le immagini in passaggio precedente. Non utilizzare `http://` oppure `https://` nel nome del registro di sistema.	stringa	`cr.astra.netapp.io` (impostazione predefinita) `example.registry.com/astra` (esempio di registro locale)

Impostazione	Utilizzare	Guida	Tipo	Esempio
`imageRegistry.secret`	Opzionale	Il nome del segreto Kubernetes utilizzato per l'autenticazione con il registro delle immagini. Il valore viene precompilato e non è richiesta alcuna azione, a meno che non sia stato configurato un registro locale e la stringa immessa per tale registro `imageRegistry.name` richiede un segreto. IMPORTANTE: Se si utilizza un registro locale che non richiede l'autorizzazione, è necessario eliminarlo `secret` linea entro `imageRegistry` in caso negativo, l'installazione non riesce.	stringa	`astra-registry-cred`

StorageClass

Impostazione	Guida	Tipo	Esempio
`storageClass`	Modificare il `storageClass` valore da `ontap-gold` A un'altra risorsa storageClass come richiesto dall'installazione. Eseguire il comando `kubectl get sc` per determinare le classi di storage configurate esistenti. Una delle classi di storage configurate per Astra Control provisioner deve essere inserita nel file manifest (`astra-control-center-<version>.manifest`) E verranno utilizzati per Astra PVS. Se non è impostata, viene utilizzata la classe di storage predefinita. NOTA: Se è configurata una classe di storage predefinita, assicurarsi che sia l'unica classe di storage con l'annotazione predefinita.	stringa	`ontap-gold`

VolumeReclaimPolicy

Impostazione	Guida	Tipo	Opzioni
`volumeReclaimPolicy`	In questo modo viene impostata la policy di recupero per il PVS di Astra. Impostare questo criterio su `Retain` Conserva i volumi persistenti dopo l'eliminazione di Astra. Impostare questo criterio su `Delete` elimina i volumi persistenti dopo l'eliminazione di astra. Se questo valore non viene impostato, il PVS viene mantenuto.	stringa	• `Retain` (Valore predefinito) • `Delete`

Tipo di ingessType

Impostazione	Guida	Tipo	Opzioni
`ingressType`	Utilizzare uno dei seguenti tipi di ingresso: **Generico** (`ingressType: "Generic"`) (Impostazione predefinita) Utilizzare questa opzione quando si utilizza un altro controller di ingresso o si preferisce utilizzare un controller di ingresso personalizzato. Dopo aver implementato Astra Control Center, è necessario configurare "controller di ingresso" Per esporre Astra Control Center con un URL. IMPORTANTE: Se si intende utilizzare una mesh di servizio con Astra Control Center, è necessario selezionare `Generic` come tipo di ingresso e configurare il proprio "controller di ingresso". **AccTraefik** (`ingressType: "AccTraefik"`) Utilizzare questa opzione quando si preferisce non configurare un controller di ingresso. In questo modo viene implementato l'Astra Control Center `traefik` Gateway come servizio di tipo Kubernetes LoadBalancer. Astra Control Center utilizza un servizio del tipo "LoadBalancer" (`svc/traefik` Nello	stringa	• `Generic` (valore predefinito) • `AccTraefik`

Dimensione scala

Impostazione	Guida	Tipo	Opzioni
`scaleSize`	Per impostazione predefinita, Astra utilizza High Availability (ha) `scaleSize` di `Medium`, Che implementa la maggior parte dei servizi in ha e implementa più repliche per la ridondanza. Con `scaleSize` come `Small`, Astra ridurrà il numero di repliche per tutti i servizi ad eccezione dei servizi essenziali per ridurre il consumo. SUGGERIMENTO: `Medium` le implementazioni sono costituite da circa 100 pod (non inclusi i carichi di lavoro transitori. 100 pod si basa su una configurazione a tre nodi master e tre nodi worker). Tenere a conoscenza dei limiti di rete per pod che potrebbero rappresentare un problema nell'ambiente, in particolare quando si prendono in considerazione scenari di disaster recovery.	stringa	• `Small` • `Medium` (Valore predefinito)

AstraResourcesScaler

Impostazione	Guida	Tipo	Opzioni
`astraResourcesScaler`	Opzioni di scalabilità per i limiti delle risorse di AstraControlCenter. Per impostazione predefinita, Astra Control Center implementa le richieste di risorse impostate per la maggior parte dei componenti all'interno di Astra. Questa configurazione consente allo stack software Astra Control Center di migliorare le prestazioni in ambienti con maggiore carico e scalabilità delle applicazioni. Tuttavia, negli scenari che utilizzano cluster di sviluppo o test più piccoli, il campo CR `astraResourcesScalar` può essere impostato su `Off`. In questo modo vengono disattivate le richieste di risorse e viene eseguita l'implementazione su cluster più piccoli.	stringa	• `Default` (Valore predefinito) • `Off`

AdditionalValues

Aggiungere i seguenti valori aggiuntivi ad Astra Control Center CR per evitare un problema noto durante l'installazione:

```
additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

crds

Le selezioni effettuate in questa sezione determinano il modo in cui Astra Control Center deve gestire i CRD.

Impostazione	Guida	Tipo	Esempio
`crds.externalCertManager`	Se si utilizza un gestore esterno dei certificati, cambiare `externalCertManager` a. `true`. L'impostazione predefinita `false` Fa in modo che Astra Control Center installi i propri CRD di gestione dei certificati durante l'installazione. I CRDS sono oggetti a livello di cluster e l'installazione potrebbe avere un impatto su altre parti del cluster. È possibile utilizzare questo indicatore per segnalare ad Astra Control Center che questi CRD verranno installati e gestiti dall'amministratore del cluster al di fuori di Astra Control Center.	Booleano	`False` (valore predefinito)
`crds.externalTraefik`	Per impostazione predefinita, Astra Control Center installerà i CRD Traefik richiesti. I CRDS sono oggetti a livello di cluster e l'installazione potrebbe avere un impatto su altre parti del cluster. È possibile utilizzare questo indicatore per segnalare ad Astra Control Center che questi CRD verranno installati e gestiti dall'amministratore del cluster al di fuori di Astra Control Center.	Booleano	`False` (valore predefinito)

Assicurarsi di aver selezionato la classe di storage e il tipo di ingresso corretti per la configurazione prima di completare l'installazione.

esempio astra_control_center.yaml

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[cr.astra.netapp.io or your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  crds:
    externalTraefik: false
    externalCertManager: false
```

Completare l'installazione dell'Astra Control Center e dell'operatore

1. Se non lo si è già fatto in un passaggio precedente, creare il `netapp-acc` namespace (o personalizzato):

   ```
   kubectl create ns [netapp-acc or custom namespace]
   ```

2. Se si utilizza una mesh di servizio con Astra Control Center, aggiungere la seguente etichetta al `netapp-acc` o namespace personalizzato:

Il tipo di ingresso (`ingressType`) deve essere impostato su `Generic` In Astra Control Center CR prima di procedere con questo comando.

```
kubectl label ns [netapp-acc or custom namespace] istio-
injection:enabled
```

3. (Consigliato) "Attivare Strict MTLS" Per la mesh di servizio Istio:

```
kubectl apply -n istio-system -f - <<EOF
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
spec:
  mtls:
    mode: STRICT
EOF
```

4. Installare Astra Control Center in `netapp-acc` spazio dei nomi (o personalizzato):

```
kubectl apply -f astra_control_center.yaml -n [netapp-acc or custom
namespace]
```

L'operatore di Astra Control Center esegue un controllo automatico dei requisiti ambientali. Mancante "requisiti" Può causare problemi di installazione o il funzionamento non corretto di Astra Control Center. Vedere sezione successiva per verificare la presenza di messaggi di avvertenza relativi al controllo automatico del sistema.

Verificare lo stato del sistema

È possibile verificare lo stato del sistema utilizzando i comandi kubectl. Se preferisci utilizzare OpenShift, puoi utilizzare comandi oc paragonabili per le fasi di verifica.

Fasi

1. Verificare che il processo di installazione non abbia prodotto messaggi di avviso relativi ai controlli di convalida:

```
kubectl get acc [astra or custom Astra Control Center CR name] -n
[netapp-acc or custom namespace] -o yaml
```

Ulteriori messaggi di avviso sono riportati anche nei registri dell'operatore di Astra Control Center.

2. Correggere eventuali problemi dell'ambiente segnalati dai controlli automatici dei requisiti.

> È possibile correggere i problemi assicurandosi che l'ambiente soddisfi i requisiti "requisiti" Per Astra Control Center.

3. Verificare che tutti i componenti del sistema siano installati correttamente.

```
kubectl get pods -n [netapp-acc or custom namespace]
```

Ogni pod deve avere uno stato di `Running`. L'implementazione dei pod di sistema potrebbe richiedere alcuni minuti.

Espandere per la risposta del campione

```
acc-helm-repo-5bd77c9ddd-8wxm2             1/1     Running     0
1h
activity-5bb474dc67-8l9ss                  1/1     Running     0
1h
activity-5bb474dc67-qbrtq                  1/1     Running     0
1h
api-token-authentication-6wbj2             1/1     Running     0
1h
api-token-authentication-9pgw6             1/1     Running     0
1h
api-token-authentication-tqf6d             1/1     Running     0
1h
asup-5495f44dbd-z4kft                      1/1     Running     0
1h
authentication-6fdd899858-5x45s            1/1     Running     0
1h
bucketservice-84d47487d-n9xgp              1/1     Running     0
1h
bucketservice-84d47487d-t5jhm              1/1     Running     0
1h
cert-manager-5dcb7648c4-hbldc              1/1     Running     0
1h
cert-manager-5dcb7648c4-nr9qf              1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-bk2tf   1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-pfnck   1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-ngz2x       1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-rwtbn       1/1     Running     0
1h
certificates-5f5b7b4dd-52tnj               1/1     Running     0
1h
certificates-5f5b7b4dd-gtjbx               1/1     Running     0
1h
certificates-expiry-check-28477260-dz5vw   0/1     Completed   0
1h
cloud-extension-6f58cc579c-lzfmv           1/1     Running     0
1h
cloud-extension-6f58cc579c-zw2km           1/1     Running     0
1h
cluster-orchestrator-79dd5c8d95-qjg92      1/1     Running     0
1h
```

```
composite-compute-85dc84579c-nz82f         1/1     Running     0
1h
composite-compute-85dc84579c-wx2z2         1/1     Running     0
1h
composite-volume-bff6f4f76-789nj           1/1     Running     0
1h
composite-volume-bff6f4f76-kwnd4           1/1     Running     0
1h
credentials-79fd64f788-m7m8f               1/1     Running     0
1h
credentials-79fd64f788-qnc6c               1/1     Running     0
1h
entitlement-f69cdbd77-4p2kn                1/1     Running     0
1h
entitlement-f69cdbd77-hswm6                1/1     Running     0
1h
features-7b9585444c-7xd7m                  1/1     Running     0
1h
features-7b9585444c-dcqwc                  1/1     Running     0
1h
fluent-bit-ds-crq8m                        1/1     Running     0
1h
fluent-bit-ds-gmgq8                        1/1     Running     0
1h
fluent-bit-ds-gzr4f                        1/1     Running     0
1h
fluent-bit-ds-j6sf6                        1/1     Running     0
1h
fluent-bit-ds-v4t9f                        1/1     Running     0
1h
fluent-bit-ds-x7j59                        1/1     Running     0
1h
graphql-server-6cc684fb46-2x8lr            1/1     Running     0
1h
graphql-server-6cc684fb46-bshbd            1/1     Running     0
1h
hybridauth-84599f79fd-fjc7k                1/1     Running     0
1h
hybridauth-84599f79fd-s9pmn                1/1     Running     0
1h
identity-95df98cb5-dvlmz                   1/1     Running     0
1h
identity-95df98cb5-krf59                   1/1     Running     0
1h
influxdb2-0                                1/1     Running     0
1h
```

```
keycloak-operator-6d4d688697-cfq8b         1/1     Running     0
1h
krakend-5d5c8f4668-7bq8g                   1/1     Running     0
1h
krakend-5d5c8f4668-t8hbn                   1/1     Running     0
1h
license-689cdd4595-2gsc8                   1/1     Running     0
1h
license-689cdd4595-g6vwk                   1/1     Running     0
1h
login-ui-57bb599956-4fwgz                  1/1     Running     0
1h
login-ui-57bb599956-rhztb                  1/1     Running     0
1h
loki-0                                     1/1     Running     0
1h
metrics-facade-846999bdd4-f7jdm            1/1     Running     0
1h
metrics-facade-846999bdd4-lnsxl            1/1     Running     0
1h
monitoring-operator-6c9d6c4b8c-ggkrl       2/2     Running     0
1h
nats-0                                     1/1     Running     0
1h
nats-1                                     1/1     Running     0
1h
nats-2                                     1/1     Running     0
1h
natssync-server-6df7d6cc68-9v2gd           1/1     Running     0
1h
nautilus-64b7fbdd98-bsgwb                  1/1     Running     0
1h
nautilus-64b7fbdd98-djlhw                  1/1     Running     0
1h
openapi-864584bccc-75nlv                   1/1     Running     0
1h
openapi-864584bccc-zh6bx                   1/1     Running     0
1h
polaris-consul-consul-server-0             1/1     Running     0
1h
polaris-consul-consul-server-1             1/1     Running     0
1h
polaris-consul-consul-server-2             1/1     Running     0
1h
polaris-keycloak-0                         1/1     Running     2 (1h
ago)      1h
```

```
polaris-keycloak-1                          1/1     Running     0
1h
polaris-keycloak-db-0                       1/1     Running     0
1h
polaris-keycloak-db-1                       1/1     Running     0
1h
polaris-keycloak-db-2                       1/1     Running     0
1h
polaris-mongodb-0                           1/1     Running     0
1h
polaris-mongodb-1                           1/1     Running     0
1h
polaris-mongodb-2                           1/1     Running     0
1h
polaris-ui-66476dcf87-f6s8j                 1/1     Running     0
1h
polaris-ui-66476dcf87-ztjk7                 1/1     Running     0
1h
polaris-vault-0                             1/1     Running     0
1h
polaris-vault-1                             1/1     Running     0
1h
polaris-vault-2                             1/1     Running     0
1h
public-metrics-bfc4fc964-x4m79              1/1     Running     0
1h
storage-backend-metrics-7dbb88d4bc-g78cj    1/1     Running     0
1h
storage-provider-5969b5df5-hjvcm            1/1     Running     0
1h
storage-provider-5969b5df5-r79ld            1/1     Running     0
1h
task-service-5fc9dc8d99-4q4f4               1/1     Running     0
1h
task-service-5fc9dc8d99-815zl               1/1     Running     0
1h
task-service-task-purge-28485735-fdzkd      1/1     Running     0
12m
telegraf-ds-2rgm4                           1/1     Running     0
1h
telegraf-ds-4qp6r                           1/1     Running     0
1h
telegraf-ds-77frs                           1/1     Running     0
1h
telegraf-ds-bc725                           1/1     Running     0
1h
```

```
telegraf-ds-cvmxf                          1/1     Running     0
1h
telegraf-ds-tqzgj                          1/1     Running     0
1h
telegraf-rs-5wtd8                          1/1     Running     0
1h
telemetry-service-6747866474-5djnc         1/1     Running     0
1h
telemetry-service-6747866474-thb7r         1/1     Running     1 (1h
ago)      1h
tenancy-5669854fb6-gzdzf                   1/1     Running     0
1h
tenancy-5669854fb6-xvsm2                   1/1     Running     0
1h
traefik-8f55f7d5d-4lgfw                    1/1     Running     0
1h
traefik-8f55f7d5d-j4wt6                    1/1     Running     0
1h
traefik-8f55f7d5d-p6gcq                    1/1     Running     0
1h
trident-svc-7cb5bb4685-54cnq               1/1     Running     0
1h
trident-svc-7cb5bb4685-b28xh               1/1     Running     0
1h
vault-controller-777b9bbf88-b5bqt          1/1     Running     0
1h
vault-controller-777b9bbf88-fdfd8          1/1     Running     0
1h
```

4. (Facoltativo) guardare `acc-operator` registri per monitorare l'avanzamento:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> `accHost` la registrazione del cluster è una delle ultime operazioni e, in caso di errore, la distribuzione non avrà esito negativo. In caso di errore di registrazione del cluster indicato nei registri, è possibile tentare di nuovo la registrazione tramite "Aggiungere il flusso di lavoro del cluster nell'interfaccia utente" O API.

5. Una volta eseguiti tutti i pod, verificare che l'installazione sia stata eseguita correttamente (`READY` è `True`) E ottieni la password di configurazione iniziale che userai quando accedi ad Astra Control Center:

```
kubectl get AstraControlCenter -n [netapp-acc or custom namespace]
```

Risposta:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  24.02.0-69
10.111.111.111  True
```

> Copiare il valore UUID. La password è `ACC-` Seguito dal valore UUID (`ACC-[UUID]` oppure, in questo esempio, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Impostare l'ingresso per il bilanciamento del carico

È possibile configurare un controller di ingresso Kubernetes che gestisce l'accesso esterno ai servizi. Queste procedure forniscono esempi di configurazione per un controller di ingresso se si utilizza il valore predefinito di `ingressType: "Generic"` Nella risorsa personalizzata di Astra Control Center (`astra_control_center.yaml`). Non è necessario utilizzare questa procedura, se specificato `ingressType: "AccTraefik"` Nella risorsa personalizzata di Astra Control Center (`astra_control_center.yaml`).

Dopo l'implementazione di Astra Control Center, è necessario configurare il controller di ingresso per esporre Astra Control Center con un URL.

Le fasi di installazione variano a seconda del tipo di controller di ingresso utilizzato. Astra Control Center supporta molti tipi di controller di ingresso. Queste procedure di configurazione forniscono alcuni esempi di passaggi per alcuni tipi di controller di ingresso comuni.

Prima di iniziare

- Il necessario "controller di ingresso" dovrebbe essere già implementato.
- Il "classe di ingresso" corrispondente al controller di ingresso dovrebbe già essere creato.

Passaggi per l'ingresso di Istio

1. Configurare l'ingresso Istio.

   > Questa procedura presuppone che Istio venga distribuito utilizzando il profilo di configurazione "predefinito".

2. Raccogliere o creare il certificato e il file della chiave privata desiderati per Ingress Gateway.

   È possibile utilizzare un certificato CA o autofirmato. Il nome comune deve essere l'indirizzo Astra (FQDN).

   Esempio di comando:

   ```
   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
   tls.crt
   ```

3. Crea un segreto `tls secret name` di tipo `kubernetes.io/tls` Per una chiave privata TLS e un certificato in `istio-system namespace` Come descritto in TLS secrets (segreti TLS).

Esempio di comando:

```
kubectl create secret tls [tls secret name] --key="tls.key"
--cert="tls.crt" -n istio-system
```

> Il nome del segreto deve corrispondere a. `spec.tls.secretName` fornito in `istio-ingress.yaml` file.

4. Implementare una risorsa di ingresso in `netapp-acc` namespace (o personalizzato) che utilizza il tipo di risorsa v1 per uno schema (`istio-Ingress.yaml` in questo esempio):

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. Applicare le modifiche:

```
kubectl apply -f istio-Ingress.yaml
```

6. Controllare lo stato dell'ingresso:

```
kubectl get ingress -n [netapp-acc or custom namespace]
```

Risposta:

```
NAME    CLASS HOSTS              ADDRESS         PORTS   AGE
ingress istio astra.example.com 172.16.103.248  80, 443 1h
```

7. Completare l'installazione di Astra Control Center.

Procedura per il controller di ingresso Nginx

1. Creare un segreto di tipo `kubernetes.io/tls` Per una chiave privata TLS e un certificato in `netapp-acc` (o con nome personalizzato) come descritto in "Segreti TLS".
2. Implementare una risorsa income in `netapp-acc` namespace (o personalizzato) che utilizza il tipo di risorsa v1 per uno schema (`nginx-Ingress.yaml` in questo esempio):

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

3. Applicare le modifiche:

```
kubectl apply -f nginx-Ingress.yaml
```

> NetApp consiglia di installare il controller nginx come implementazione piuttosto che come `daemonSet`.

Procedura per il controller di ingresso OpenShift

1. Procurarsi il certificato e ottenere la chiave, il certificato e i file CA pronti per l'uso con il percorso OpenShift.
2. Creare il percorso OpenShift:

```
oc create route edge --service=traefik --port=web -n [netapp-acc or
custom namespace] --insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

Accedere all'interfaccia utente di Astra Control Center

Dopo aver installato Astra Control Center, modificherai la password per l'amministratore predefinito ed effettuerai l'accesso alla dashboard dell'interfaccia utente di Astra Control Center.

Fasi

1. In un browser, immettere l'FQDN (compreso il `https://` prefisso) utilizzato in `astraAddress` in `astra_control_center.yaml` CR quando Astra Control Center è stato installato.
2. Accettare i certificati autofirmati, se richiesto.

   > È possibile creare un certificato personalizzato dopo l'accesso.

3. Nella pagina di accesso di Astra Control Center, inserire il valore utilizzato per `email` poll `astra_control_center.yaml` CR quando Astra Control Center è stato installato, seguito dalla password di configurazione iniziale (`ACC-[UUID]`).

   > Se si immette una password errata per tre volte, l'account admin viene bloccato per 15 minuti.

4. Selezionare **Login**.
5. Modificare la password quando richiesto.

   > Se si tratta del primo accesso e si dimentica la password e non sono stati ancora creati altri account utente amministrativi, contattare "Supporto NetApp" per assistenza per il recupero della password.

6. (Facoltativo) rimuovere il certificato TLS autofirmato esistente e sostituirlo con un "Certificato TLS personalizzato firmato da un'autorità di certificazione (CA)".

Risolvere i problemi di installazione

Se uno dei servizi è in `Error` stato, è possibile esaminare i registri. Cercare i codici di risposta API nell'intervallo da 400 a 500. Questi indicano il luogo in cui si è verificato un guasto.

Opzioni

- Per esaminare i registri dell'operatore di Astra Control Center, immettere quanto segue:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

- Per controllare l'output di Astra Control Center CR:

```
kubectl get acc -n [netapp-acc or custom namespace] -o yaml
```

Procedure di installazione alternative

- **Installa con Red Hat OpenShift OperatorHub**: USA questo "procedura alternativa" Per installare Astra Control Center su OpenShift utilizzando OperatorHub.
- **Installare nel cloud pubblico con backend Cloud Volumes ONTAP**: Utilizzare "queste procedure" Per installare Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure con un backend di storage Cloud Volumes ONTAP.

Cosa succederà

- (Opzionale) a seconda dell'ambiente, completare la post-installazione "fasi di configurazione".
- "Dopo aver installato Astra Control Center, effettuato l'accesso all'interfaccia utente e modificato la password, è necessario impostare una licenza, aggiungere cluster, abilitare l'autenticazione, gestire lo storage e aggiungere bucket".

Configurare un gestore esterno dei certificati

Se nel cluster Kubernetes esiste già un cert manager, è necessario eseguire alcuni passaggi preliminari in modo che Astra Control Center non installi il proprio cert manager.

Fasi

1. Verificare che sia installato un gestore dei certificati:

```
kubectl get pods -A | grep 'cert-manager'
```

   Esempio di risposta:

```
cert-manager   essential-cert-manager-84446f49d5-sf2zd              1/1
Running    0      6d5h
cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
Running    0      6d5h
cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq      1/1
Running    0      6d5h
```

2. Creare una coppia certificato/chiave per `astraAddress` FQDN:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt
```

Esempio di risposta:

```
Generating a 2048 bit RSA private key
..................+++
.........................................+++
writing new private key to 'tls.key'
```

3. Creare un segreto con i file generati in precedenza:

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n
<cert-manager-namespace>
```

Esempio di risposta:

```
secret/selfsigned-tls created
```

4. Creare un `ClusterIssuer` file che è **esattamente** il seguente, ma include la posizione dello spazio dei nomi in cui si trova il `cert-manager` i pod sono installati:

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Esempio di risposta:

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. Verificare che il `ClusterIssuer` è venuto in su correttamente. `Ready` deve essere `True` prima di procedere:

```
kubectl get ClusterIssuer
```

Esempio di risposta:

```
NAME                      READY   AGE
astra-ca-clusterissuer    True    9s
```

6. Completare il "Processo di installazione di Astra Control Center". Esiste un "Fase di configurazione richiesta per il cluster Astra Control Center YAML" In cui si modifica il valore CRD per indicare che il gestore dei certificati è installato esternamente. È necessario completare questa fase durante l'installazione in modo che Astra Control Center riconosca il cert manager esterno.

Installare Astra Control Center utilizzando OpenShift OperatorHub

Se utilizzi Red Hat OpenShift, puoi installare Astra Control Center usando l'operatore certificato Red Hat. Seguire questa procedura per installare Astra Control Center da "Catalogo Red Hat Ecosystem" Oppure utilizzando Red Hat OpenShift Container Platform.

Una volta completata questa procedura, tornare alla procedura di installazione per completare la "fasi rimanenti" per verificare che l'installazione sia riuscita e accedere.

Prima di iniziare

- **Soddisfare i requisiti ambientali**: "Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center".

> Implementare Astra Control Center in un terzo dominio di errore o in un sito secondario. Questa opzione è consigliata per la replica delle applicazioni e il disaster recovery perfetto.

- **Assicurare operatori di cluster e servizi API sani**:
  - Dal cluster OpenShift, assicurati che tutti gli operatori del cluster siano in buono stato:

```
oc get clusteroperators
```

- Dal cluster OpenShift, assicurati che tutti i servizi API siano in buono stato:

```
oc get apiservices
```

- **Assicurarsi che un FQDN instradabile**: Il FQDN Astra che si intende utilizzare può essere instradato al cluster. Ciò significa che si dispone di una voce DNS nel server DNS interno o si sta utilizzando un percorso URL principale già registrato.
- **Ottieni autorizzazioni OpenShift**: Avrai bisogno di tutte le autorizzazioni necessarie e dell'accesso a Red Hat OpenShift Container Platform per eseguire i passaggi di installazione descritti.
- **Configura un cert manager**: Se nel cluster esiste già un cert manager, è necessario eseguirne alcuni "fasi preliminari" In modo che Astra Control Center non installi il proprio cert manager. Per impostazione predefinita, Astra Control Center installa il proprio cert manager durante l'installazione.
- **Configura controller ingresso Kubernetes**: Se si dispone di un controller ingresso Kubernetes che gestisce l'accesso esterno a servizi, come il bilanciamento del carico in un cluster, è necessario configurarlo per l'utilizzo con Astra Control Center:
  a. Creare lo spazio dei nomi dell'operatore:

```
oc create namespace netapp-acc-operator
```

  b. "Completare la configurazione" per il proprio tipo di controller di ingresso.
- * (Solo driver SAN ONTAP) Abilita multipath*: Se stai utilizzando un driver SAN ONTAP, assicurati che multipath sia abilitato su tutti i tuoi cluster Kubernetes.

È inoltre necessario considerare quanto segue:

- **Ottenere l'accesso al Registro di sistema dell'immagine di controllo Astra di NetApp**:

  È possibile ottenere le immagini di installazione e i miglioramenti delle funzionalità per Astra Control, come Astra Control provisioner, dal registro delle immagini di NetApp.

  a. Registrare l'ID dell'account Astra Control necessario per accedere al Registro di sistema.

     Puoi visualizzare l'ID dell'account nell'interfaccia utente Web di Astra Control Service. Selezionare l'icona a forma di figura in alto a destra nella pagina, selezionare **accesso API** e annotare l'ID account.

  b. Nella stessa pagina, selezionare **generate API token**, copiare la stringa del token API negli Appunti e salvarla nell'editor.
  c. Accedere al registro Astra Control:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

- **Installare una mesh di servizio per comunicazioni sicure**: Si consiglia vivamente di proteggere i canali di comunicazione del cluster host Astra Control utilizzando un "mesh di servizio supportata".

> L'integrazione di Astra Control Center con una mesh di servizio può essere eseguita solo durante Astra Control Center "installazione" e non indipendente da questo processo. Il passaggio da un ambiente con mesh a un ambiente senza mesh non è supportato.

Per l'uso della mesh del servizio Istio, è necessario effettuare le seguenti operazioni:

- Aggiungere un `istio-injection:enabled` Etichetta nel namespace Astra prima di implementare Astra Control Center.
- Utilizzare `Generic` impostazione ingresso e fornire un ingresso alternativo per "bilanciamento del carico esterno".
- Per i cluster Red Hat OpenShift, è necessario definire `NetworkAttachmentDefinition` Su tutti i namespace Astra Control Center associati (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` per i cluster di applicazioni o qualsiasi namespace personalizzato che sia stato sostituito).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Fasi

- Scarica ed estrai Astra Control Center
- Completare ulteriori passaggi se si utilizza un registro locale
- Individuare la pagina di installazione dell'operatore
- Installare l'operatore
- Installare Astra Control Center

> Non eliminare l'operatore di Astra Control Center (ad esempio, `kubectl delete -f astra_control_center_operator_deploy.yaml`) In qualsiasi momento durante l'installazione o il funzionamento di Astra Control Center per evitare di eliminare i pod.

Scarica ed estrai Astra Control Center

Scarica le immagini di Astra Control Center da una delle seguenti posizioni:

- **Registro di sistema dell'immagine del servizio di controllo Astra**: Utilizzare questa opzione se non si utilizza un registro locale con le immagini del centro di controllo Astra o se si preferisce questo metodo per il download del pacchetto dal sito di supporto NetApp.
- **Sito di supporto NetApp**: Utilizzare questa opzione se si utilizza un registro locale con le immagini del Centro di controllo Astra.

Registro delle immagini di Astra Control

1. Effettua l'accesso ad Astra Control Service.
2. Nella Dashboard, selezionare **distribuire un'istanza autogestita di Astra Control**.
3. Seguire le istruzioni per accedere al registro delle immagini di Astra Control, estrarre l'immagine di installazione di Astra Control Center ed estrarre l'immagine.

Sito di supporto NetApp

1. Scarica il bundle contenente Astra Control Center (`astra-control-center-[version].tar.gz`) da "Pagina di download di Astra Control Center".
2. (Consigliato ma opzionale) Scarica il bundle di certificati e firme per Astra Control Center (`astra-control-center-certs-[version].tar.gz`) per verificare la firma del bundle.

   ```
   tar -vxzf astra-control-center-certs-[version].tar.gz
   ```

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig astra-
   control-center-[version].tar.gz
   ```

   Viene visualizzato l'output `Verified OK` una volta completata la verifica.

3. Estrarre le immagini dal bundle Astra Control Center:

   ```
   tar -vxzf astra-control-center-[version].tar.gz
   ```

Completare ulteriori passaggi se si utilizza un registro locale

Se si intende inviare il pacchetto Astra Control Center al registro locale, è necessario utilizzare il plugin della riga di comando di NetApp Astra kubectl.

Installare il plug-in NetApp Astra kubectl

Completare questi passaggi per installare il più recente plugin della riga di comando di NetApp Astra kubectl.

Prima di iniziare

NetApp fornisce binari per plug-in per diverse architetture CPU e sistemi operativi. Prima di eseguire questa attività, è necessario conoscere la CPU e il sistema operativo in uso.

Se il plug-in è già stato installato da un'installazione precedente, "assicurarsi di disporre della versione più recente" prima di completare questa procedura.

Fasi

1. Elencare i binari del plugin NetApp Astra kubectl disponibili e annotare il nome del file necessario per il sistema operativo e l'architettura della CPU:

   La libreria di plugin kubectl fa parte del bundle tar e viene estratta nella cartella `kubectl-astra`.

   ```
   ls kubectl-astra/
   ```

2. Spostare il binario corretto nel percorso corrente e rinominarlo `kubectl-astra`:

   ```
   cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
   ```

Aggiungere le immagini al registro

1. Se si prevede di inviare il pacchetto Astra Control Center al registro locale, completare la sequenza di passaggi appropriata per il motore del contenitore:

Docker

a. Passare alla directory root del tarball. Viene visualizzata la `acc.manifest.bundle.yaml` file e queste directory:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Trasferire le immagini del pacchetto nella directory delle immagini di Astra Control Center nel registro locale. Eseguire le seguenti sostituzioni prima di eseguire `push-images` comando:

- Sostituire <BUNDLE_FILE> con il nome del file bundle di controllo Astra (`acc.manifest.bundle.yaml`).
- Sostituire <MY_FULL_REGISTRY_PATH> con l’URL del repository Docker; ad esempio, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
- Sostituire <MY_REGISTRY_USER> con il nome utente.
- Sostituire <MY_REGISTRY_TOKEN> con un token autorizzato per il registro.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

a. Passare alla directory root del tarball. Vengono visualizzati il file e la directory seguenti:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Accedere al Registro di sistema:

```
podman login <YOUR_REGISTRY>
```

c. Preparare ed eseguire uno dei seguenti script personalizzato per la versione di Podman utilizzata. Sostituire <MY_FULL_REGISTRY_PATH> con l'URL del repository che include le sottodirectory.

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

Il percorso dell'immagine creato dallo script deve essere simile al seguente, a seconda della configurazione del Registro di sistema:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Modificare la directory:

```
cd manifests
```

Individuare la pagina di installazione dell’operatore

1. Completare una delle seguenti procedure per accedere alla pagina di installazione dell’operatore:

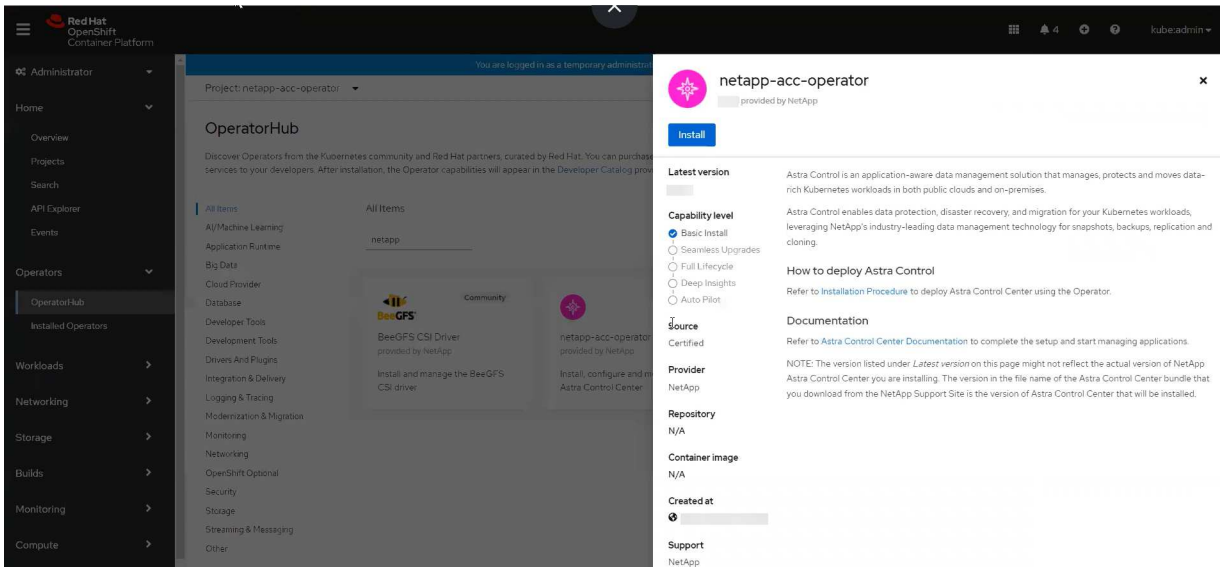
Console Web Red Hat OpenShift

a. Accedere all'interfaccia utente di OpenShift Container Platform.

b. Dal menu laterale, selezionare **Operator (operatori) > OperatorHub**.

> Con questo operatore è possibile eseguire l'aggiornamento solo alla versione corrente di Astra Control Center.

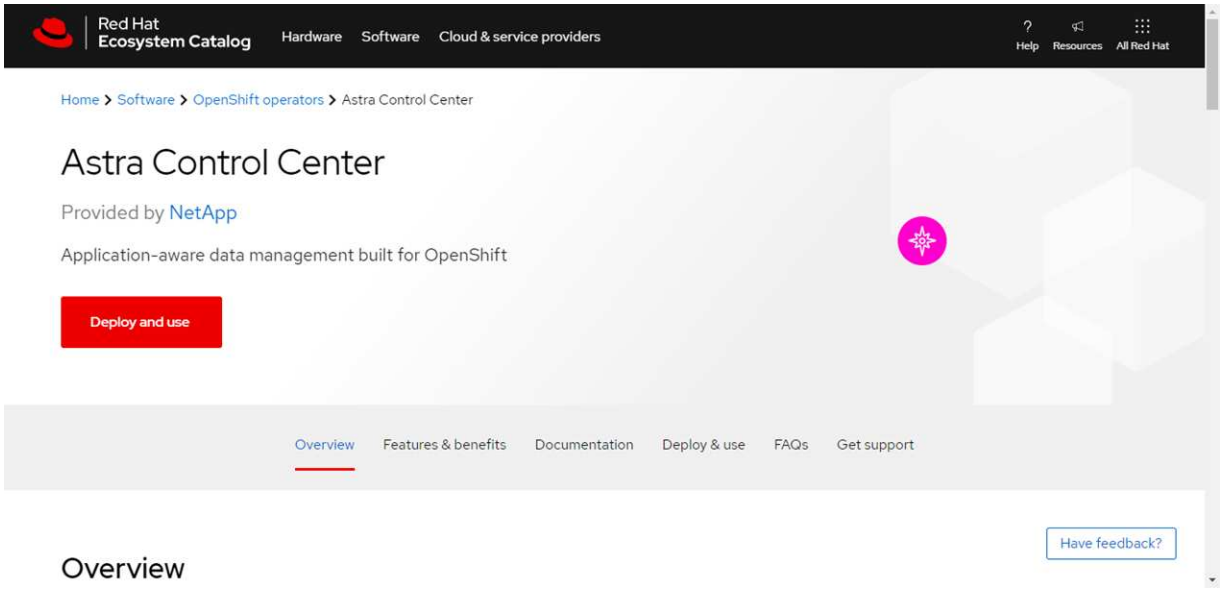
c. Cercare `netapp-acc` E selezionare l'operatore NetApp Astra Control Center.



Catalogo Red Hat Ecosystem

a. Selezionare NetApp Astra Control Center "operatore".

b. Selezionare **Deploy and Use** (distribuzione e utilizzo).



Installare l'operatore

1. Completare la pagina **Install Operator** (Installazione operatore) e installare l'operatore:

   > L'operatore sarà disponibile in tutti gli spazi dei nomi dei cluster.

   a. Selezionare lo spazio dei nomi dell'operatore o. `netapp-acc-operator` lo spazio dei nomi verrà creato automaticamente come parte dell'installazione dell'operatore.

   b. Selezionare una strategia di approvazione manuale o automatica.

   > Si consiglia l'approvazione manuale. Per ogni cluster dovrebbe essere in esecuzione una sola istanza dell'operatore.

   c. Selezionare **Installa**.

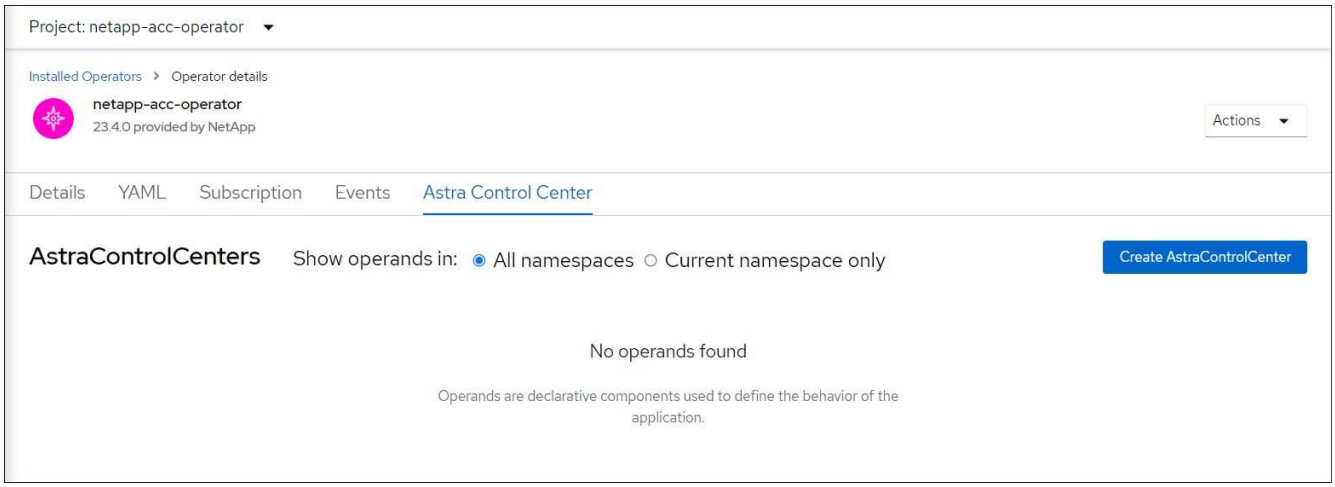
   > Se è stata selezionata una strategia di approvazione manuale, verrà richiesto di approvare il piano di installazione manuale per questo operatore.

2. Dalla console, accedere al menu OperatorHub e verificare che l'installazione dell'operatore sia stata eseguita correttamente.

Installare Astra Control Center

1. Dalla console all'interno della scheda **Astra Control Center** dell'operatore Astra Control Center, selezionare **Create AstraControlCenter**

   .



2. Completare il `Create AstraControlCenter` campo del modulo:

   a. Mantenere o regolare il nome di Astra Control Center.

   b. Aggiungere etichette per Astra Control Center.

   c. Attiva o disattiva il supporto automatico. Si consiglia di mantenere la funzionalità di supporto automatico.

   d. Inserire il nome FQDN o l'indirizzo IP di Astra Control Center. Non entrare `http://` oppure `https://` nel campo dell'indirizzo.

   e. Immettere la versione di Astra Control Center, ad esempio 24.02.0-69.

f. Immettere un nome account, un indirizzo e-mail e un cognome amministratore.

g. Scegliere una policy di recupero dei volumi di `Retain`, `Recycle`, o. `Delete`. Il valore predefinito è `Retain`.

h. Selezionare la dimensione della scala dell'installazione.

> Per impostazione predefinita, Astra utilizza High Availability (ha) `scaleSize` di `Medium`, Che implementa la maggior parte dei servizi in ha e implementa più repliche per la ridondanza. Con `scaleSize` come `Small`, Astra ridurrà il numero di repliche per tutti i servizi ad eccezione dei servizi essenziali per ridurre il consumo.

i. selezionare il tipo di ingresso:

- **Generico** (`ingressType: "Generic"`) (Impostazione predefinita)

  Utilizzare questa opzione quando si utilizza un altro controller di ingresso o si preferisce utilizzare un controller di ingresso personalizzato. Dopo aver implementato Astra Control Center, è necessario configurare "controller di ingresso" Per esporre Astra Control Center con un URL.

- **AccTraefik** (`ingressType: "AccTraefik"`)

  Utilizzare questa opzione quando si preferisce non configurare un controller di ingresso. In questo modo viene implementato l'Astra Control Center `traefik` Gateway come servizio di tipo Kubernetes "LoadBalancer".

Astra Control Center utilizza un servizio del tipo "LoadBalancer" (`svc/traefik` Nello spazio dei nomi di Astra Control Center) e richiede l'assegnazione di un indirizzo IP esterno accessibile. Se nel proprio ambiente sono consentiti i bilanciatori di carico e non ne è già configurato uno, è possibile utilizzare MetalLB o un altro servizio di bilanciamento del carico esterno per assegnare un indirizzo IP esterno al servizio. Nella configurazione del server DNS interno, puntare il nome DNS scelto per Astra Control Center sull'indirizzo IP con bilanciamento del carico.

> Per ulteriori informazioni sul tipo di servizio "LoadBalancer" e sull'ingresso, fare riferimento a. "Requisiti".

a. In **Registro immagini**, utilizzare il valore predefinito a meno che non sia stato configurato un registro locale. Per un registro locale, sostituire questo valore con il percorso del Registro di sistema dell'immagine locale in cui sono state inserite le immagini in un passaggio precedente. Non entrare `http://` oppure `https://` nel campo dell'indirizzo.

b. Se si utilizza un registro di immagini che richiede l'autenticazione, inserire il segreto dell'immagine.

> Se si utilizza un registro che richiede l'autenticazione, creare un segreto sul cluster.

c. Inserire il nome admin.

d. Configurare la scalabilità delle risorse.

e. Fornire la classe di storage predefinita.

> Se è configurata una classe di storage predefinita, assicurarsi che sia l'unica classe di storage con l'annotazione predefinita.

f. Definire le preferenze di gestione CRD.

3. Selezionare la vista YAML per rivedere le impostazioni selezionate.
4. Selezionare `Create`.

Creare un segreto di registro

Se si utilizza un registro che richiede l'autenticazione, creare un segreto nel cluster OpenShift e immettere il nome segreto nel `Create AstraControlCenter` campo del modulo.

1. Creare uno spazio dei nomi per l'operatore Astra Control Center:

```
oc create ns [netapp-acc-operator or custom namespace]
```

2. Creare un segreto in questo namespace:

```
oc create secret docker-registry astra-registry-cred -n [netapp-acc-
operator or custom namespace] --docker-server=[your_registry_path]
--docker username=[username] --docker-password=[token]
```

> Astra Control supporta solo i segreti del Registro di sistema di Docker.

3. Completare i campi rimanenti in Il campo Create AstraControlCenter Form (Crea modulo AstraControlCenter).

Cosa succederà

Completare il "fasi rimanenti" Per verificare che Astra Control Center sia stato installato correttamente, configurare un controller di ingresso (opzionale) e accedere all'interfaccia utente. Inoltre, sarà necessario eseguire "attività di installazione" al termine dell'installazione.

Installare il centro di controllo Astra con un backend di storage Cloud Volumes ONTAP

Con Astra Control Center, puoi gestire le tue app in un ambiente di cloud ibrido con cluster Kubernetes e istanze di Cloud Volumes ONTAP autogestiti. Puoi implementare Astra Control Center nei tuoi cluster Kubernetes on-premise o in uno dei cluster Kubernetes autogestiti nell'ambiente cloud.

Con una di queste implementazioni, è possibile eseguire operazioni di gestione dei dati delle applicazioni utilizzando Cloud Volumes ONTAP come back-end dello storage. È inoltre possibile configurare un bucket S3 come destinazione del backup.

Per installare Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) e Microsoft Azure con un backend di storage Cloud Volumes ONTAP, eseguire i seguenti passaggi a seconda dell'ambiente cloud in uso.

- Implementare Astra Control Center in Amazon Web Services

- Implementare Astra Control Center nella piattaforma Google Cloud
- Implementare Astra Control Center in Microsoft Azure

Puoi gestire le tue applicazioni nelle distribuzioni con cluster Kubernetes autogestiti, come OpenShift Container Platform (OCP). Solo i cluster OCP autogestiti sono validati per l'implementazione di Astra Control Center.

Implementare Astra Control Center in Amazon Web Services

Puoi implementare Astra Control Center su un cluster Kubernetes autogestito ospitato su un cloud pubblico Amazon Web Services (AWS).

Ciò di cui hai bisogno per AWS

Prima di implementare Astra Control Center in AWS, sono necessari i seguenti elementi:

- Licenza Astra Control Center. Fare riferimento a. "Requisiti di licenza di Astra Control Center".
- "Soddisfare i requisiti di Astra Control Center".
- Account NetApp Cloud Central
- Se si utilizza OCP, autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per creare i pod)
- Credenziali AWS, Access ID e Secret Key con autorizzazioni che consentono di creare bucket e connettori
- Accesso e login al Registro dei container elastici (ECR) dell'account AWS
- Per accedere all'interfaccia utente di Astra Control è richiesta la zona ospitata di AWS e la voce Amazon Route 53

Requisiti dell'ambiente operativo per AWS

Astra Control Center richiede il seguente ambiente operativo per AWS:

- Red Hat OpenShift Container Platform dalla versione 4.11 alla 4.13

Assicurarsi che l'ambiente operativo scelto per ospitare Astra Control Center soddisfi i requisiti delle risorse di base descritti nella documentazione ufficiale dell'ambiente.

Astra Control Center richiede risorse specifiche oltre ai requisiti delle risorse dell'ambiente. Fare riferimento a. "Requisiti dell'ambiente operativo di Astra Control Center".

> Il token di registro AWS scade tra 12 ore, dopodiché sarà necessario rinnovare la password di registro dell'immagine di Docker.

Panoramica dell'implementazione per AWS

Di seguito viene fornita una panoramica del processo di installazione di Astra Control Center per AWS con Cloud Volumes ONTAP come backend di storage.

Ciascuna di queste fasi viene illustrata più dettagliatamente di seguito.

1. Assicurarsi di disporre di autorizzazioni IAM sufficienti.
2. Installare un cluster RedHat OpenShift su AWS.

3. Configurare AWS.
4. Configurare NetApp BlueXP per AWS.
5. Installare Astra Control Center per AWS.

Assicurarsi di disporre di autorizzazioni IAM sufficienti

Assicurarsi di disporre di ruoli e autorizzazioni IAM sufficienti per installare un cluster RedHat OpenShift e un connettore NetApp BlueXP (in precedenza Cloud Manager).

Vedere "Credenziali AWS iniziali".

Installare un cluster RedHat OpenShift su AWS

Installare un cluster RedHat OpenShift Container Platform su AWS.

Per istruzioni sull'installazione, vedere "Installazione di un cluster su AWS in OpenShift Container Platform".

Configurare AWS

Quindi, configurare AWS per creare una rete virtuale, configurare istanze di calcolo EC2 e creare un bucket AWS S3. Se non si riesce ad accedere al registro delle immagini del Centro di controllo Astra di NetApp, è necessario anche creare un ECR (Elastic Container Registry) per ospitare le immagini del Centro di controllo Astra e inviare le immagini al Registro di sistema.

Seguire la documentazione di AWS per completare i seguenti passaggi. Vedere "Documentazione di installazione di AWS".

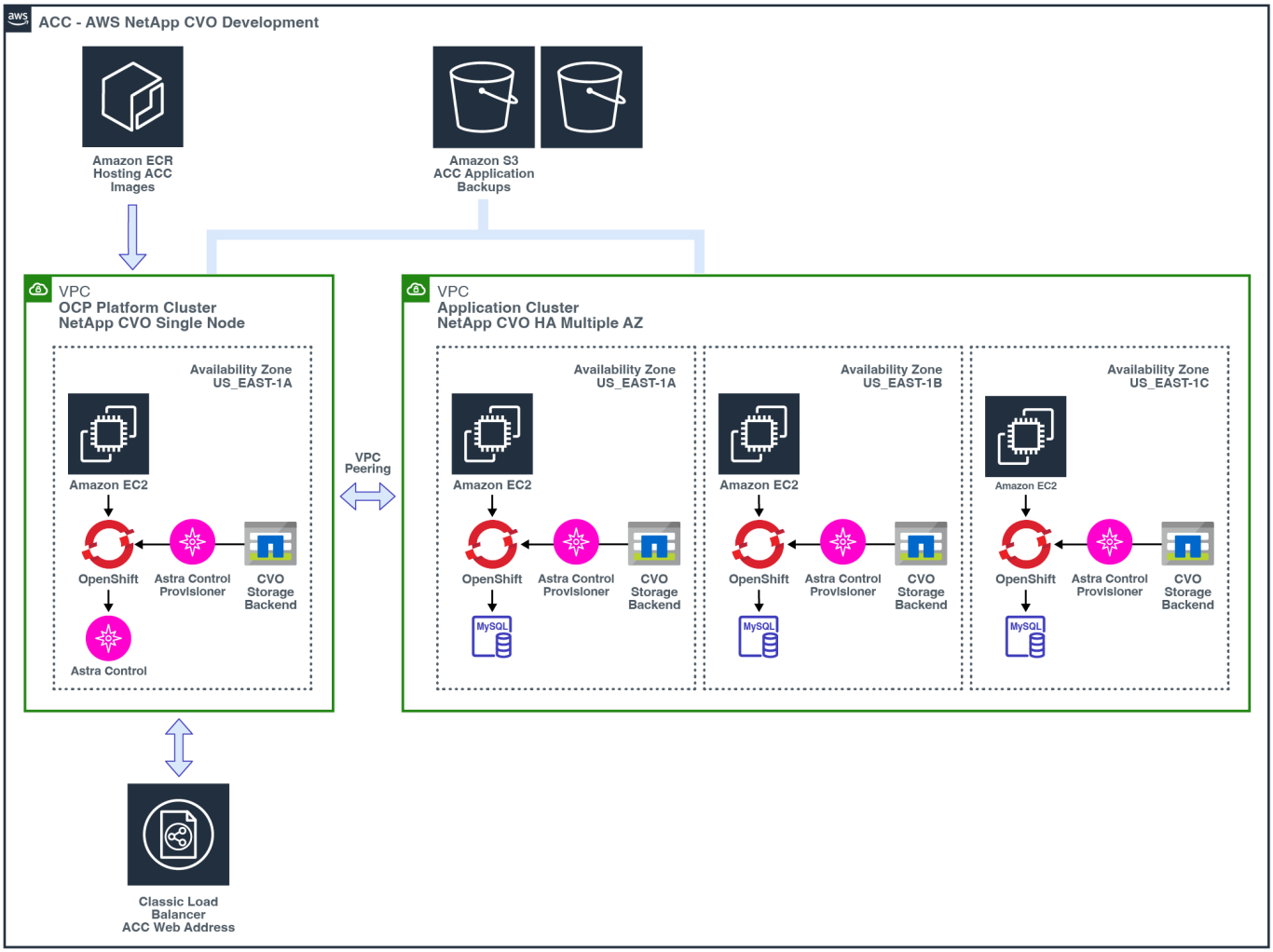
1. Creare una rete virtuale AWS.
2. Esaminare le istanze di calcolo EC2. Può trattarsi di un server bare metal o di macchine virtuali in AWS.
3. Se il tipo di istanza non corrisponde già ai requisiti minimi di risorsa Astra per i nodi master e worker, modificare il tipo di istanza in AWS per soddisfare i requisiti Astra. Fare riferimento a. "Requisiti di Astra Control Center".
4. Creare almeno un bucket AWS S3 per memorizzare i backup.
5. (Facoltativo) se non è possibile accedere al registro delle immagini di NetApp, procedere come segue:
   a. Creare un AWS Elastic Container Registry (ECR) per ospitare tutte le immagini di Astra Control Center.

      > Se non si crea ECR, il centro di controllo Astra non può accedere ai dati di monitoraggio da un cluster contenente Cloud Volumes ONTAP con un backend AWS. Il problema si verifica quando il cluster che si tenta di rilevare e gestire utilizzando Astra Control Center non dispone dell'accesso ad AWS ECR.

   b. Trasferire le immagini di Astra Control Center nel registro definito.

      > Il token AWS Elastic Container Registry (ECR) scade dopo 12 ore e causa il fallimento delle operazioni di cloni tra cluster. Questo problema si verifica quando si gestisce un backend di storage da Cloud Volumes ONTAP configurato per AWS. Per correggere questo problema, autenticare nuovamente con ECR e generare un nuovo segreto per la ripresa delle operazioni di clonazione.

Ecco un esempio di implementazione di AWS:



Configurare NetApp BlueXP per AWS

Utilizzando NetApp BlueXP (in precedenza Cloud Manager), creare uno spazio di lavoro, aggiungere un connettore ad AWS, creare un ambiente di lavoro e importare il cluster.

Seguire la documentazione di BlueXP per completare i seguenti passaggi. Vedere quanto segue:

- "Introduzione a Cloud Volumes ONTAP in AWS".
- "Creare un connettore in AWS utilizzando BlueXP"

Fasi

1. Aggiungi le tue credenziali a BlueXP.
2. Creare un'area di lavoro.
3. Aggiungere un connettore per AWS. Scegliere AWS come provider.
4. Crea un ambiente di lavoro per il tuo ambiente cloud.
   a. Location: "Amazon Web Services (AWS)"
   b. Tipo: "Cloud Volumes ONTAP ha"
5. Importare il cluster OpenShift. Il cluster si connetterà all'ambiente di lavoro appena creato.
   a. Per visualizzare i dettagli del cluster NetApp, selezionare **K8s** > **elenco cluster** > **Dettagli cluster**.

b. Nell'angolo in alto a destra, osserva la versione di Astra Control Provisioner.

c. Si noti che le classi di storage cluster Cloud Volumes ONTAP mostrano NetApp come provider.

   In questo modo, il cluster Red Hat OpenShift viene importato e viene assegnata una classe di storage predefinita. Selezionare la classe di storage.
   Astra Control provisioner viene installato automaticamente nell'ambito del processo di importazione e rilevamento.

6. Tenere presenti tutti i volumi e i volumi persistenti in questa implementazione di Cloud Volumes ONTAP.

> Cloud Volumes ONTAP può funzionare come nodo singolo o in alta disponibilità. Se ha è attivato, annotare lo stato ha e lo stato di implementazione del nodo in esecuzione in AWS.

Installare Astra Control Center per AWS

Seguire lo standard "Istruzioni di installazione di Astra Control Center".

> AWS utilizza il tipo di bucket S3 generico.

Implementare Astra Control Center nella piattaforma Google Cloud

Puoi implementare Astra Control Center su un cluster Kubernetes autogestito ospitato su un cloud pubblico Google Cloud Platform (GCP).

Cosa ti serve per GCP

Prima di implementare Astra Control Center in GCP, sono necessari i seguenti elementi:

- Licenza Astra Control Center. Fare riferimento a. "Requisiti di licenza di Astra Control Center".
- "Soddisfare i requisiti di Astra Control Center".
- Account NetApp Cloud Central
- Se si utilizza OCP, Red Hat OpenShift Container Platform (OCP) da 4.11 a 4.13
- Se si utilizza OCP, autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per creare i pod)
- GCP Service account con autorizzazioni che consentono di creare bucket e connettori

Requisiti dell'ambiente operativo per GCP

Assicurarsi che l'ambiente operativo scelto per ospitare Astra Control Center soddisfi i requisiti delle risorse di base descritti nella documentazione ufficiale dell'ambiente.

Astra Control Center richiede risorse specifiche oltre ai requisiti delle risorse dell'ambiente. Fare riferimento a. "Requisiti dell'ambiente operativo di Astra Control Center".

Panoramica dell'implementazione per GCP

Di seguito viene fornita una panoramica del processo di installazione di Astra Control Center su un cluster OCP autogestiti in GCP con Cloud Volumes ONTAP come backend di storage.

Ciascuna di queste fasi viene illustrata più dettagliatamente di seguito.

1. Installare un cluster RedHat OpenShift su GCP.
2. Crea un progetto GCP e un cloud privato virtuale.
3. Assicurarsi di disporre di autorizzazioni IAM sufficienti.
4. Configurare GCP.
5. Configurare NetApp BlueXP per GCP.
6. Installare Astra Control Center per GCP.

Installare un cluster RedHat OpenShift su GCP

Il primo passo consiste nell'installare un cluster RedHat OpenShift su GCP.

Per istruzioni sull'installazione, consultare quanto segue:

- "Installazione di un cluster OpenShift in GCP"
- "Creazione di un account di servizio GCP"

Crea un progetto GCP e un cloud privato virtuale

Creare almeno un progetto GCP e Virtual Private Cloud (VPC).

> OpenShift potrebbe creare i propri gruppi di risorse. Inoltre, è necessario definire un VPC GCP. Fare riferimento alla documentazione di OpenShift.

È possibile creare un gruppo di risorse del cluster di piattaforme e un gruppo di risorse del cluster OpenShift dell'applicazione di destinazione.

Assicurarsi di disporre di autorizzazioni IAM sufficienti

Assicurarsi di disporre di ruoli e autorizzazioni IAM sufficienti per installare un cluster RedHat OpenShift e un connettore NetApp BlueXP (in precedenza Cloud Manager).

Vedere "Credenziali e permessi GCP iniziali".

Configurare GCP

Quindi, configurare GCP per creare un VPC, configurare istanze di calcolo e creare un Google Cloud Object Storage. Se non è possibile accedere al registro delle immagini di NetApp Astra Control Center, è necessario creare un registro dei contenitori di Google per ospitare le immagini di Astra Control Center e inviare le immagini a questo registro.

Seguire la documentazione GCP per completare i seguenti passaggi. Vedere Installazione del cluster OpenShift in GCP.

1. Creare un progetto GCP e un VPC nel GCP che si intende utilizzare per il cluster OCP con backend CVO.
2. Esaminare le istanze di calcolo. Questo può essere un server bare metal o VM in GCP.
3. Se il tipo di istanza non corrisponde già ai requisiti minimi di risorsa Astra per i nodi master e worker, modificare il tipo di istanza in GCP per soddisfare i requisiti Astra. Fare riferimento a. "Requisiti di Astra Control Center".
4. Crea almeno un bucket di storage cloud GCP per memorizzare i tuoi backup.
5. Creare un segreto, necessario per l'accesso al bucket.

6. (Facoltativo) se non è possibile accedere al registro delle immagini di NetApp, procedere come segue:

   a. Creare un Google Container Registry per ospitare le immagini di Astra Control Center.

   b. Impostare l'accesso al Google Container Registry per il push/pull di Docker per tutte le immagini di Astra Control Center.

      Esempio: Le immagini di Astra Control Center possono essere inviate a questo registro inserendo il seguente script:

      ```
      gcloud auth activate-service-account <service account email address>
      --key-file=<GCP Service Account JSON file>
      ```

      Questo script richiede un file manifesto di Astra Control Center e la posizione del Google Image Registry. Esempio:

```
manifestfile=acc.manifest.bundle.yaml
GCP_CR_REGISTRY=<target GCP image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
    docker push $GCP_CR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

7. Impostare le zone DNS.

Configurare NetApp BlueXP per GCP

Utilizzando NetApp BlueXP (in precedenza Cloud Manager), creare uno spazio di lavoro, aggiungere un connettore a GCP, creare un ambiente di lavoro e importare il cluster.

Seguire la documentazione di BlueXP per completare i seguenti passaggi. Vedere "Introduzione a Cloud Volumes ONTAP in GCP".

Prima di iniziare

- Accesso all'account di servizio GCP con i ruoli e le autorizzazioni IAM richiesti

Fasi

1. Aggiungi le tue credenziali a BlueXP. Vedere "Aggiunta di account GCP".
2. Aggiungere un connettore per GCP.
   a. Scegliere "GCP" come provider.
   b. Immettere le credenziali GCP. Vedere "Creazione di un connettore in GCP da BlueXP".

c. Assicurarsi che il connettore sia in funzione e passare a tale connettore.

3. Crea un ambiente di lavoro per il tuo ambiente cloud.
   a. Location: Italy
   b. Tipo: "Cloud Volumes ONTAP ha"
4. Importare il cluster OpenShift. Il cluster si connetterà all'ambiente di lavoro appena creato.
   a. Per visualizzare i dettagli del cluster NetApp, selezionare **K8s** > **elenco cluster** > **Dettagli cluster**.
   b. Nell'angolo in alto a destra, osserva la versione di Astra Control Provisioner.
   c. Si noti che le classi di storage del cluster Cloud Volumes ONTAP mostrano "NetApp" come provider.

      In questo modo, il cluster Red Hat OpenShift viene importato e viene assegnata una classe di storage predefinita. Selezionare la classe di storage.
      Astra Control provisioner viene installato automaticamente nell'ambito del processo di importazione e rilevamento.
5. Tenere presenti tutti i volumi e i volumi persistenti in questa implementazione di Cloud Volumes ONTAP.

> Cloud Volumes ONTAP può operare come un singolo nodo o in alta disponibilità (ha). Se ha è attivato, annotare lo stato ha e lo stato di implementazione del nodo in esecuzione in GCP.

Installare Astra Control Center per GCP

Seguire lo standard "Istruzioni di installazione di Astra Control Center".

> GCP utilizza il tipo di bucket S3 generico.

1. Generare il Docker Secret per estrarre le immagini per l'installazione di Astra Control Center:

```
kubectl create secret docker-registry <secret name> --docker
-server=<Registry location> --docker-username=_json_key --docker
-password="$(cat <GCP Service Account JSON file>)" --namespace=pcloud
```

Implementare Astra Control Center in Microsoft Azure

Puoi implementare Astra Control Center su un cluster Kubernetes autogestito ospitato su un cloud pubblico Microsoft Azure.

Ciò di cui hai bisogno per Azure

Prima di implementare Astra Control Center in Azure, sono necessari i seguenti elementi:

- Licenza Astra Control Center. Fare riferimento a. "Requisiti di licenza di Astra Control Center".
- "Soddisfare i requisiti di Astra Control Center".
- Account NetApp Cloud Central
- Se si utilizza OCP, Red Hat OpenShift Container Platform (OCP) da 4.11 a 4.13
- Se si utilizza OCP, autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per creare i pod)

- Credenziali Azure con autorizzazioni che consentono di creare bucket e connettori

Requisiti dell'ambiente operativo per Azure

Assicurarsi che l'ambiente operativo scelto per ospitare Astra Control Center soddisfi i requisiti delle risorse di base descritti nella documentazione ufficiale dell'ambiente.

Astra Control Center richiede risorse specifiche oltre ai requisiti delle risorse dell'ambiente. Fare riferimento a. "Requisiti dell'ambiente operativo di Astra Control Center".

Panoramica dell'implementazione di Azure

Ecco una panoramica del processo di installazione di Astra Control Center per Azure.

Ciascuna di queste fasi viene illustrata più dettagliatamente di seguito.

1. Installare un cluster RedHat OpenShift su Azure.
2. Creare gruppi di risorse Azure.
3. Assicurarsi di disporre di autorizzazioni IAM sufficienti.
4. Configurare Azure.
5. Configurare NetApp BlueXP (in precedenza Cloud Manager) per Azure.
6. Installare e configurare Astra Control Center per Azure.

Installare un cluster RedHat OpenShift su Azure

Il primo passo consiste nell'installare un cluster RedHat OpenShift su Azure.

Per istruzioni sull'installazione, consultare quanto segue:

- "Installazione del cluster OpenShift su Azure".
- "Installazione di un account Azure".

Creare gruppi di risorse Azure

Creare almeno un gruppo di risorse Azure.

> OpenShift potrebbe creare i propri gruppi di risorse. Oltre a questi, è necessario definire anche i gruppi di risorse di Azure. Fare riferimento alla documentazione di OpenShift.

È possibile creare un gruppo di risorse del cluster di piattaforme e un gruppo di risorse del cluster OpenShift dell'applicazione di destinazione.

Assicurarsi di disporre di autorizzazioni IAM sufficienti

Assicurarsi di disporre di ruoli e autorizzazioni IAM sufficienti per l'installazione di un cluster RedHat OpenShift e di un connettore NetApp BlueXP.

Vedere "Credenziali e permessi di Azure".

Configurare Azure

Quindi, configurare Azure per creare una rete virtuale, configurare istanze di calcolo e creare un container

Azure Blob. Se non è possibile accedere al registro delle immagini del Centro di controllo Astra di NetApp, è necessario creare anche un ACR (Azure Container Registry) per ospitare le immagini del Centro di controllo Astra e inviare le immagini al Registro di sistema.

Seguire la documentazione di Azure per completare i seguenti passaggi. Vedere "Installazione del cluster OpenShift su Azure".

1. Creare una rete virtuale Azure.
2. Esaminare le istanze di calcolo. Si tratta di un server bare metal o di macchine virtuali in Azure.
3. Se il tipo di istanza non corrisponde già ai requisiti minimi di risorsa Astra per i nodi master e worker, modificare il tipo di istanza in Azure per soddisfare i requisiti Astra. Fare riferimento a. "Requisiti di Astra Control Center".
4. Creare almeno un container Azure Blob per memorizzare i backup.
5. Creare un account storage. Ti servirà un account di storage per creare un container da utilizzare come bucket in Astra Control Center.
6. Creare un segreto, necessario per l'accesso al bucket.
7. (Facoltativo) se non è possibile accedere al registro delle immagini di NetApp, procedere come segue:

   a. Creare un Azure Container Registry (ACR) per ospitare le immagini di Astra Control Center.
   b. Impostare l'accesso ACR per la funzione push/pull di Docker per tutte le immagini di Astra Control Center.
   c. Inviare le immagini di Astra Control Center a questo registro utilizzando il seguente script:

      ```
      az acr login -n <AZ ACR URL/Location>
      This script requires the Astra Control Center manifest file and your
      Azure ACR location.
      ```

      Esempio:

```
manifestfile=acc.manifest.bundle.yaml
AZ_ACR_REGISTRY=<target Azure ACR image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

8. Impostare le zone DNS.

Configurare NetApp BlueXP (in precedenza Cloud Manager) per Azure

Utilizzando BlueXP (in precedenza Cloud Manager), creare un'area di lavoro, aggiungere un connettore ad Azure, creare un ambiente di lavoro e importare il cluster.

Seguire la documentazione di BlueXP per completare i seguenti passaggi. Vedere "Introduzione a BlueXP in Azure".

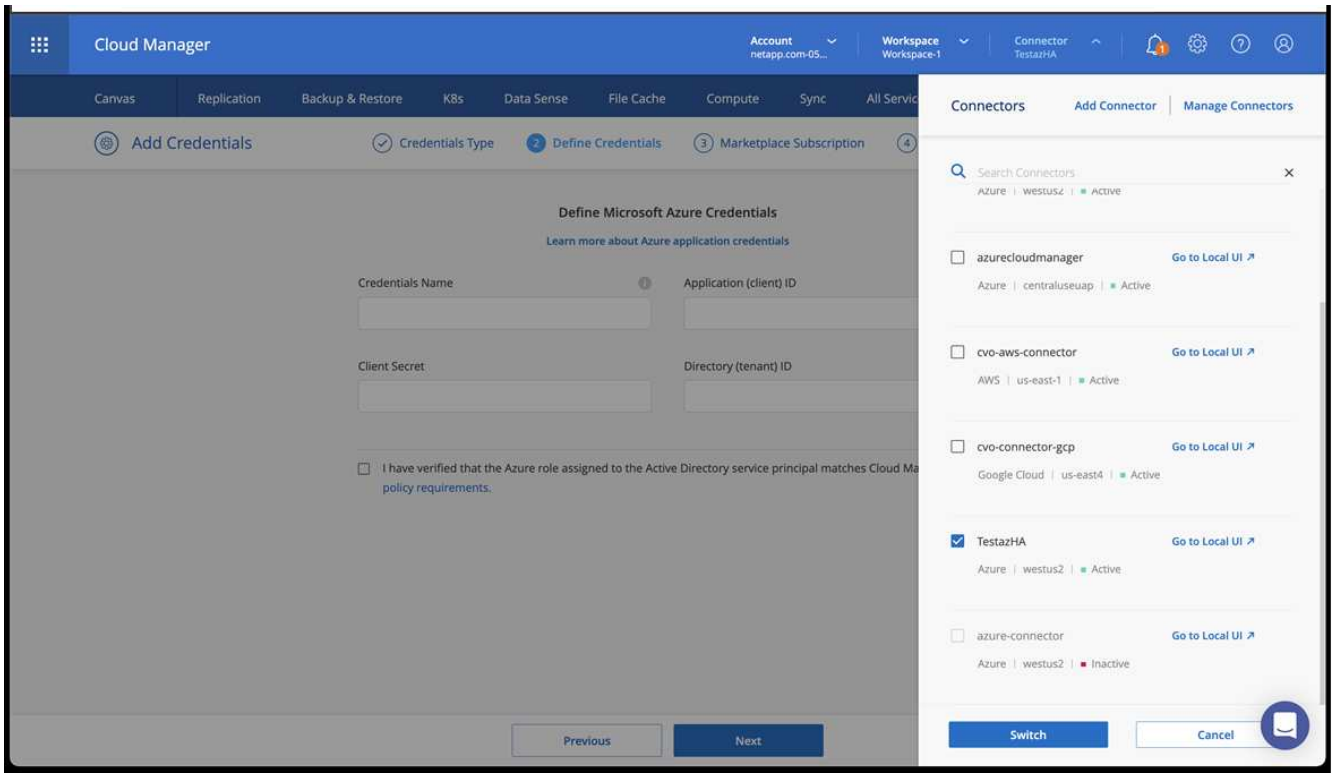
Prima di iniziare

Accesso all'account Azure con le autorizzazioni e i ruoli IAM richiesti

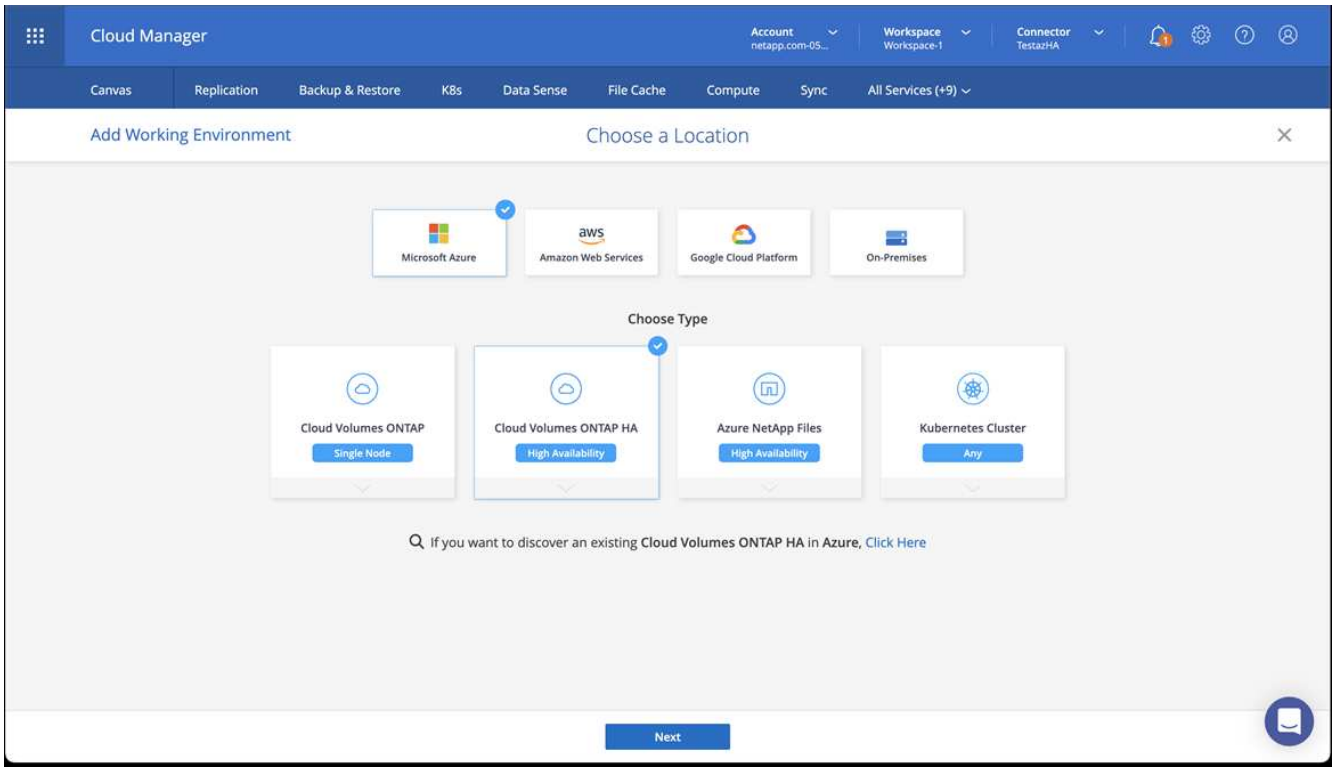
Fasi

1. Aggiungi le tue credenziali a BlueXP.
2. Aggiungere un connettore per Azure. Vedere "Policy BlueXP".
   a. Scegliere **Azure** come provider.
   b. Immettere le credenziali Azure, inclusi ID applicazione, segreto client e ID directory (tenant).

      Vedere "Creazione di un connettore in Azure da BlueXPr".
3. Assicurarsi che il connettore sia in funzione e passare a tale connettore.

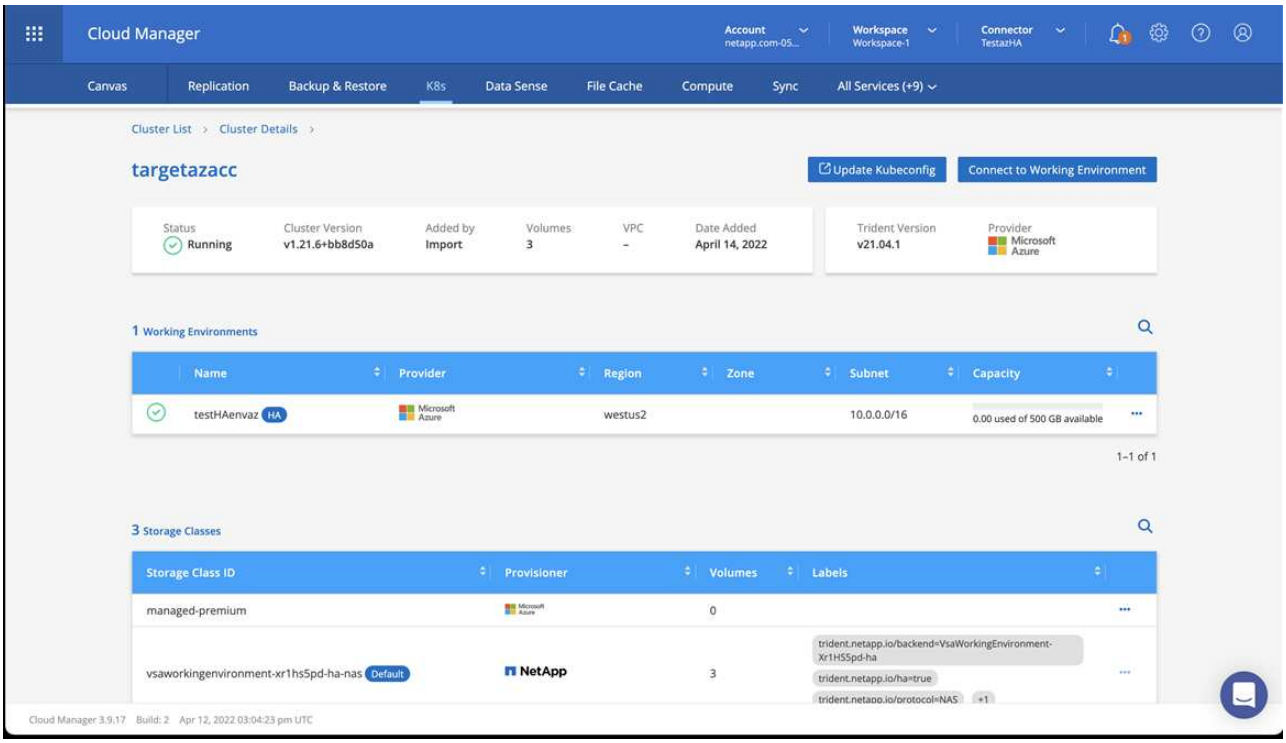


4. Crea un ambiente di lavoro per il tuo ambiente cloud.
   a. Percorso: "Microsoft Azure".
   b. Tipo: "Cloud Volumes ONTAP ha".



5. Importare il cluster OpenShift. Il cluster si connetterà all'ambiente di lavoro appena creato.

   a. Per visualizzare i dettagli del cluster NetApp, selezionare **K8s** > **elenco cluster** > **Dettagli cluster**.



   b. Nell'angolo in alto a destra, osserva la versione di Astra Control Provisioner.

   c. Si noti che le classi di storage cluster Cloud Volumes ONTAP mostrano NetApp come provider.

In questo modo viene importato il cluster Red Hat OpenShift e viene assegnata una classe di storage predefinita. Selezionare la classe di storage.

Astra Control provisioner viene installato automaticamente nell'ambito del processo di importazione e rilevamento.

6. Tenere presenti tutti i volumi e i volumi persistenti in questa implementazione di Cloud Volumes ONTAP.
7. Cloud Volumes ONTAP può funzionare come nodo singolo o in alta disponibilità. Se ha è attivato, annotare lo stato ha e lo stato di implementazione del nodo in esecuzione in Azure.

Installare e configurare Astra Control Center per Azure

Installare Astra Control Center con lo standard "istruzioni per l'installazione".

Utilizzando Astra Control Center, aggiungere un bucket Azure. Fare riferimento a. "Configurare Astra Control Center e aggiungere i bucket".

Configurare Astra Control Center dopo l'installazione

A seconda dell'ambiente in uso, potrebbe essere necessaria una configurazione aggiuntiva dopo l'installazione di Astra Control Center.

Rimuovere le limitazioni delle risorse

Alcuni ambienti utilizzano gli oggetti ResourceQuotas e LimitRanges per impedire alle risorse di uno spazio dei nomi di consumare tutta la CPU e la memoria disponibili nel cluster. Astra Control Center non imposta limiti massimi, pertanto non sarà conforme a tali risorse. Se l'ambiente è configurato in questo modo, è necessario rimuovere tali risorse dagli spazi dei nomi in cui si intende installare Astra Control Center.

Per recuperare e rimuovere le quote e i limiti, procedere come segue. In questi esempi, l'output del comando viene visualizzato immediatamente dopo il comando.

Fasi

1. Ottenere le quote delle risorse in `netapp-acc` namespace (o personalizzato):

   ```
   kubectl get quota -n [netapp-acc or custom namespace]
   ```

   Risposta:

   ```
   NAME          AGE   REQUEST                                          LIMIT
   pods-high     16s   requests.cpu: 0/20, requests.memory: 0/100Gi
   limits.cpu: 0/200, limits.memory: 0/1000Gi
   pods-low      15s   requests.cpu: 0/1, requests.memory: 0/1Gi
   limits.cpu: 0/2, limits.memory: 0/2Gi
   pods-medium   16s   requests.cpu: 0/10, requests.memory: 0/20Gi
   limits.cpu: 0/20, limits.memory: 0/200Gi
   ```

2. Eliminare tutte le quote delle risorse in base al nome:

```
kubectl delete resourcequota pods-high -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-low -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-medium -n [netapp-acc or custom
namespace]
```

3. Ottenere gli intervalli di limite in `netapp-acc` namespace (o personalizzato):

```
kubectl get limits -n [netapp-acc or custom namespace]
```

Risposta:

```
NAME              CREATED AT
cpu-limit-range   2022-06-27T19:01:23Z
```

4. Eliminare gli intervalli di limiti in base al nome:

```
kubectl delete limitrange cpu-limit-range -n [netapp-acc or custom
namespace]
```

Aggiungere un certificato TLS personalizzato

Astra Control Center utilizza per impostazione predefinita un certificato TLS autofirmato per il traffico dei controller di ingresso (solo in alcune configurazioni) e l'autenticazione dell'interfaccia utente Web con i browser Web. Per l'utilizzo in produzione, è necessario rimuovere il certificato TLS autofirmato esistente e sostituirlo con un certificato TLS firmato da un'autorità di certificazione (CA).

> Il certificato autofirmato predefinito viene utilizzato per due tipi di connessione:
>
> - Connessioni HTTPS all'interfaccia utente Web di Astra Control Center
> - Traffico del controller di ingresso (solo se `ingressType: "AccTraefik"` la proprietà è stata impostata in `astra_control_center.yaml` Durante l'installazione di Astra Control Center)
>
> La sostituzione del certificato TLS predefinito sostituisce il certificato utilizzato per l'autenticazione di queste connessioni.

Prima di iniziare

- Kubernetes cluster con Astra Control Center installato
- Accesso amministrativo a una shell dei comandi sul cluster da eseguire `kubectl` comandi
- Chiave privata e file di certificato dalla CA

Rimuovere il certificato autofirmato

Rimuovere il certificato TLS autofirmato esistente.

1. Utilizzando SSH, accedere al cluster Kubernetes che ospita Astra Control Center come utente amministrativo.
2. Individuare il segreto TLS associato al certificato corrente utilizzando il seguente comando, sostituendo `<ACC-deployment-namespace>` Con lo spazio dei nomi di implementazione di Astra Control Center:

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. Eliminare il certificato e il segreto attualmente installati utilizzando i seguenti comandi:

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
```

```
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

Aggiungere un nuovo certificato utilizzando la riga di comando

Aggiungere un nuovo certificato TLS firmato da una CA.

1. Utilizzare il seguente comando per creare il nuovo segreto TLS con la chiave privata e i file di certificato della CA, sostituendo gli argomenti tra parentesi <> con le informazioni appropriate:

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. Utilizzare il seguente comando e l'esempio per modificare il file CRD (Custom Resource Definition) del cluster e modificare `spec.selfSigned` valore a. `spec.ca.secretName` Per fare riferimento al segreto TLS creato in precedenza:

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
```

   CRD:

```
#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. Utilizzare il seguente comando e l'output di esempio per confermare che le modifiche sono corrette e che il cluster è pronto per validare i certificati, sostituendo `<ACC-deployment-namespace>` Con lo spazio dei nomi di implementazione di Astra Control Center:

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
```

Risposta:

```
Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. Creare il `certificate.yaml` file utilizzando il seguente esempio, sostituendo i valori segnaposto tra parentesi <> con le informazioni appropriate:

> In questo esempio viene utilizzato il `dnsNames` Per specificare l'indirizzo DNS di Astra Control Center. Astra Control Center non supporta l'utilizzo della proprietà Common Name (CN) per specificare l'indirizzo DNS.

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  <strong>name: <certificate-name></strong>
  namespace: <ACC-deployment-namespace>
spec:
  <strong>secretName: <certificate-secret-name></strong>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  <strong>- <astra.dnsname.example.com></strong> #Replace with the
correct Astra Control Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. Creare il certificato utilizzando il seguente comando:

```
kubectl apply -f certificate.yaml
```

6. Utilizzando il seguente comando e l'output di esempio, verificare che il certificato sia stato creato correttamente e con gli argomenti specificati durante la creazione (ad esempio nome, durata, scadenza di rinnovo e nomi DNS).

```
kubectl describe certificate -n <ACC-deployment-namespace>
```

Risposta:

```
Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. Modificare il CRD degli archivi TLS in modo che punti al nuovo nome segreto del certificato utilizzando il seguente comando ed esempio, sostituendo i valori segnaposto tra parentesi <> con le informazioni appropriate

```
kubectl edit tlsstores.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
spec:
  defaultCertificate:
    secretName: <certificate-secret-name>
```

8. Modificare l'opzione TLS CRD di ingresso per indicare il nuovo segreto del certificato utilizzando il seguente comando ed esempio, sostituendo i valori segnaposto tra parentesi <> con le informazioni appropriate:

```
kubectl edit ingressroutes.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
 tls:
    secretName: <certificate-secret-name>
```

9. Utilizzando un browser Web, accedere all'indirizzo IP di implementazione di Astra Control Center.
10. Verificare che i dettagli del certificato corrispondano ai dettagli del certificato installato.
11. Esportare il certificato e importare il risultato nel gestore dei certificati nel browser Web.

Configurare Astra Control Center

Aggiungere una licenza per Astra Control Center

Quando si installa Astra Control Center, è già installata una licenza di valutazione integrata. Se stai valutando Astra Control Center, puoi saltare questo passaggio.

È possibile aggiungere una nuova licenza utilizzando l'interfaccia utente di Astra Control o. "API di controllo Astra".

Le licenze di Astra Control Center misurano le risorse CPU utilizzando le unità CPU di Kubernetes e tengono conto delle risorse CPU assegnate ai nodi di lavoro di tutti i cluster Kubernetes gestiti. Le licenze si basano sull'utilizzo di vCPU. Per ulteriori informazioni sul calcolo delle licenze, fare riferimento a. "Licensing".

> Se l'installazione supera il numero concesso in licenza di unità CPU, Astra Control Center impedisce la gestione di nuove applicazioni. Quando viene superata la capacità, viene visualizzato un avviso.

> Per aggiornare una licenza di valutazione o una licenza completa, fare riferimento a. "Aggiornare una licenza esistente".

Prima di iniziare

- Accesso a un'istanza di Astra Control Center appena installata.
- Autorizzazioni per il ruolo di amministratore.
- R "File di licenza NetApp" (NLF).

Fasi

1. Accedere all'interfaccia utente di Astra Control Center.
2. Selezionare **account** > **licenza**.
3. Selezionare **Aggiungi licenza**.
4. Individuare il file di licenza (NLF) scaricato.
5. Selezionare **Aggiungi licenza**.

La pagina **account** > **licenza** visualizza le informazioni sulla licenza, la data di scadenza, il numero di serie della licenza, l'ID account e le unità CPU utilizzate.

Se si dispone di una licenza di valutazione e non si inviano dati a AutoSupport, assicurarsi di memorizzare l'ID account per evitare la perdita di dati in caso di guasto del centro di controllo Astra.

Abilita Astra Control Provisioner

Astra Trident le versioni 23,10 e successive includono la possibilità di utilizzare Astra Control Provivisioner, che consente agli utenti dotati di licenza Astra Control di accedere a funzionalità avanzate di provisioning dello storage. Astra Control Provisioner fornisce questa funzionalità estesa oltre alle funzionalità standard basate su CSI Astra Trident.

In arrivo gli update di Astra Control, Astra Control Provivisioner sostituirà Astra Trident come provisioner di storage e orchestrator e sarà obbligatorio per l'utilizzo di Astra Control. Per questo motivo, si consiglia vivamente agli utenti di Astra Control di attivare Astra Control Provisioner. Astra Trident continuerà a rimanere open source e ad essere rilasciato, mantenuto, supportato e aggiornato con le nuove CSI e altre funzionalità di NetApp.

A proposito di questa attività

È necessario seguire questa procedura se si è un utente di Astra Control Center con licenza e si sta cercando di utilizzare la funzionalità di Astra Control Provisioner. Devi seguire questa procedura anche se sei un utente di Astra Trident e desideri utilizzare le funzionalità aggiuntive fornite da Astra Control Provisioner senza utilizzare Astra Control.

Per ogni caso, la funzionalità di provisioning non è abilitata per impostazione predefinita in Astra Trident 24,02 e deve essere abilitata.

Prima di iniziare

Se stai abilitando Astra Control provisioner, esegui prima quanto segue:

Utenti di Astra Control Provisioners con Astra Control Center

- **Ottenere una licenza Astra Control Center**: È necessario un "Licenza Astra Control Center" Per abilitare Astra Control Provisioner e accedere alle funzionalità fornite.
- **Installa o esegui l'aggiornamento ad Astra Control Center 23,10 o versione successiva**: Se intendi utilizzare la funzionalità più recente di Astra Control Center (24,02) 24,02 con Astra Control.
- **Confirmi che il tuo cluster ha un'architettura di sistema AMD64**: L'immagine Astra Control Provivisioner è fornita in entrambe le architetture CPU AMD64 e ARM64, ma solo AMD64 è supportato da Astra Control Center.
- **Ottenere un account del Servizio di controllo Astra per l'accesso al Registro di sistema**: Se si intende utilizzare il Registro di sistema di controllo Astra piuttosto che il Sito di supporto NetApp per scaricare l'immagine del revisioner di controllo Astra, completare la registrazione per un "Account Astra Control Service". Dopo aver completato e inviato il modulo e creato un account BlueXP, riceverai un'email di benvenuto con Astra Control Service.
- **Se Astra Trident è installato, conferma che la sua versione si trovi all'interno di una finestra a quattro release**: Puoi eseguire un aggiornamento diretto a Astra Trident 24,02 con Astra Control Provisioner se il tuo Astra Trident si trova all'interno di una finestra a quattro release della versione 24,02. Ad esempio, puoi eseguire l'upgrade direttamente da Astra Trident 23,04 a 24,02.

Solo utenti di Astra Control provisioner

- **Ottenere una licenza Astra Control Center**: È necessario un "Licenza Astra Control Center" Per abilitare Astra Control Provisioner e accedere alle funzionalità fornite.
- **Se Astra Trident è installato, conferma che la sua versione si trovi all'interno di una finestra a quattro release**: Puoi eseguire un aggiornamento diretto a Astra Trident 24,02 con Astra Control Provisioner se il tuo Astra Trident si trova all'interno di una finestra a quattro release della versione 24,02. Ad esempio, puoi eseguire l'upgrade direttamente da Astra Trident 23,04 a 24,02.
- **Prendi un account Astra Control Service per l'accesso al Registro di sistema**: Per scaricare le immagini di Astra Control provisioner, è necessario accedere al Registro di sistema. Per iniziare, completa la registrazione per un "Account Astra Control Service". Dopo aver completato e inviato il modulo e creato un account BlueXP, riceverai un'email di benvenuto con Astra Control Service.

(Fase 1) ottenere l'immagine di Astra Control provisioner

Gli utenti di Astra Control Center possono ottenere l'immagine di Astra Control Provisioner utilizzando il Registro di sistema di Astra Control o il metodo del sito di supporto di NetApp. Gli utenti di Astra Trident che desiderano utilizzare Astra Control Protivisioner senza Astra Control devono utilizzare il metodo del Registro di sistema.

Registro delle immagini di Astra Control

> È possibile utilizzare Podman invece di Docker per i comandi di questa procedura. Se si utilizza un ambiente Windows, si consiglia di utilizzare PowerShell.

1. Accedere al Registro di sistema dell'immagine di controllo Astra di NetApp:
   a. Accedere all'interfaccia utente Web di Astra Control Service e selezionare l'icona raffigurata in alto a destra nella pagina.
   b. Selezionare **API access**.
   c. Annotare l'ID account.
   d. Nella stessa pagina, selezionare **generate API token**, copiare la stringa del token API negli Appunti e salvarla nell'editor.
   e. Accedere al registro Astra Control utilizzando il metodo preferito:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

```
crane auth login cr.astra.netapp.io -u <account-id> -p <api-
token>
```

2. (Solo registri personalizzati) attenersi alla seguente procedura per spostare l'immagine nel registro personalizzato. Se non si utilizza un registro, seguire i passaggi dell'operatore Trident nel "sezione successiva".
   a. Estrarre l'immagine di Astra Control provisioner dal Registro di sistema:

> L'immagine estratta non supporta più piattaforme e supporta solo la stessa piattaforma dell'host che ha estratto l'immagine, ad esempio Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

Esempio:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0 --platform
linux/amd64
```

   a. Contrassegnare l'immagine:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

b. Inviare l'immagine al registro personalizzato:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

> È possibile utilizzare la copia di Crane come alternativa all'esecuzione dei seguenti comandi di Docker:
> `crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0`

Sito di supporto NetApp

1. Scarica il bundle Astra Control Provisioner (`trident-acp-[version].tar`) da "Pagina di download di Astra Control Center".
2. (Consigliato ma facoltativo) scaricate il pacchetto di certificati e firme per Astra Control Center (astra-control-center-certs-[version].tar.gz) per verificare la firma del pacchetto trident-acp-[version] tar.

```
tar -vxzf astra-control-center-certs-[version].tar.gz
```

```
openssl dgst -sha256 -verify certs/AstraControlCenterDockerImages-public.pub -signature certs/trident-acp-[version].tar.sig trident-acp-[version].tar
```

3. Caricare l'immagine di Astra Control provisioner:

```
docker load < trident-acp-24.02.0.tar
```

Risposta:

```
Loaded image: trident-acp:24.02.0-linux-amd64
```

4. Contrassegnare l'immagine:

```
docker tag trident-acp:24.02.0-linux-amd64 <my_custom_registry>/trident-acp:24.02.0
```

5. Inviare l'immagine al registro personalizzato:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

(Fase 2) attiva Astra Control Provisioner in Astra Trident

Determinare se il metodo di installazione originale ha utilizzato un "Operatore (manualmente o con Helm) o tridentctl" e completare i passaggi appropriati in base al metodo originale.

Operatore Astra Trident

1. "Scaricare il programma di installazione di Astra Trident ed estrarlo".
2. Completa questi passaggi se non hai ancora installato Astra Trident o se hai rimosso l'operatore dall'implementazione originale di Astra Trident:
   a. Creare il CRD:

   ```
   kubectl create -f
   deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.y
   aml
   ```

   b. Creare lo spazio dei nomi tridente (`kubectl create namespace trident`) o confermare che lo spazio dei nomi tridente esiste ancora (`kubectl get all -n trident`). Se lo spazio dei nomi è stato rimosso, crearlo di nuovo.
3. Aggiorna Astra Trident alla versione 24.02.0:

   > Per i cluster che eseguono Kubernetes 1.24 o versioni precedenti, utilizzare `bundle_pre_1_25.yaml`. Per i cluster che eseguono Kubernetes 1.25 o versioni successive, utilizzare `bundle_post_1_25.yaml`.

   ```
   kubectl -n trident apply -f trident-installer/deploy/<bundle-
   name.yaml>
   ```

4. Verificare che Astra Trident sia in esecuzione:

   ```
   kubectl get torc -n trident
   ```

   Risposta:

   ```
   NAME      AGE
   trident   21m
   ```

5. se si dispone di un registro che utilizza segreti, creare un segreto da utilizzare per estrarre l'immagine di Astra Control Provisioner:

   ```
   kubectl create secret docker-registry <secret_name> -n trident
   --docker-server=<my_custom_registry> --docker-username=<username>
   --docker-password=<token>
   ```

6. Modificare il TridentOrchestrator CR e apportare le seguenti modifiche:

```
kubectl edit torc trident -n trident
```

a. Impostare una posizione del Registro di sistema personalizzata per l'immagine Astra Trident o estrarla dal Registro di sistema Astra Control (`tridentImage: <my_custom_registry>/trident:24.02.0` oppure `tridentImage: netapp/trident:24.02.0`).

b. Abilita Astra Control Provisioner (`enableACP: true`).

c. Impostare la posizione del Registro di sistema personalizzata per l'immagine Astra Control Provivioner o estrarla dal Registro di sistema Astra Control (`acpImage: <my_custom_registry>/trident-acp:24.02.0` oppure `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).

d. Se stabilito segreti di estrazione delle immagini in precedenza, è possibile impostarle qui (`imagePullSecrets: - <secret_name>`). Usare lo stesso nome segreto che hai stabilito nei passaggi precedenti.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
  - <secret_name>
```

7. Salvare e uscire dal file. Il processo di distribuzione si avvia automaticamente.
8. Verificare che l'operatore, la distribuzione e i replicaset siano stati creati.

```
kubectl get all -n trident
```

> In un cluster Kubernetes dovrebbe esserci solo **un'istanza** dell'operatore. Non creare implementazioni multiple dell'operatore Astra Trident.

9. Verificare `trident-acp` il container è in esecuzione e così `acpVersion` è `24.02.0` con stato di `Installed`:

```
kubectl get torc -o yaml
```

Risposta:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

tridentctl

1. "Scaricare il programma di installazione di Astra Trident ed estrarlo".
2. "Se disponi già di un Astra Trident, disinstallarlo dal cluster che lo ospita".
3. Installa Astra Trident con Astra Control Provisioner abilitato (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

4. Confermare che Astra Control Provisioner è stato abilitato:

```
./tridentctl -n trident version
```

Risposta:

```
+----------------+----------------+-------------+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +----------------+----------------
+-------------+ | 24.02.0 | 24.02.0 | 24.02.0. | +----------------
+----------------+-------------+
```

Timone

1. Se hai installato Astra Trident 23.07.1 o una versione precedente, "disinstallazione" l'operatore e gli altri componenti.
2. Se il cluster Kubernetes esegue la versione 1,24 o precedente, elimina psp:

```
kubectl delete psp tridentoperatorpod
```

3. Aggiungere il repository Astra Trident Helm:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

4. Aggiornare il grafico Helm:

```
helm repo update netapp-trident
```

Risposta:

```
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. □Happy Helming!□
```

5. Elencare le immagini:

```
./tridentctl images -n trident
```

Risposta:

```
| v1.28.0            | netapp/trident:24.02.0|
|                    | docker.io/netapp/trident-autosupport:24.02|
|                    | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                    | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                    | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                    | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                    | registry.k8s.io/sig-storage/csi-node-driver-
registrar:v2.10.0 |
|                    | netapp/trident-operator:24.02.0 (optional)
```

6. Assicurarsi che l'operatore di tridente 24.02.0 sia disponibile:

```
helm search repo netapp-trident/trident-operator --versions
```

Risposta:

```
NAME                            CHART VERSION   APP VERSION
DESCRIPTION
netapp-trident/trident-operator 100.2402.0      24.02.0         A
```

7. Utilizzare `helm install` ed eseguire una delle seguenti opzioni che includono queste impostazioni:
   - Un nome per la posizione di distribuzione
   - La versione di Astra Trident
   - Il nome dell'immagine di Astra Control provisioner
   - Il flag per abilitare il provisioner
   - (Facoltativo) percorso del Registro di sistema locale. Se si utilizza un registro locale, il "Immagini Trident" Può trovarsi in un registro o in registri diversi, ma tutte le immagini CSI devono trovarsi nello stesso registro.
   - Il namespace Trident

Opzioni

- Immagini senza registro

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-acp:24.02.0
--set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Immagini in uno o più registri

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

È possibile utilizzare `helm list` per rivedere i dettagli dell'installazione, ad esempio nome, spazio dei nomi, grafico, stato, versione dell'applicazione, e numero di revisione.

Se hai problemi nell'implementazione di Trident utilizzando Helm, esegui questo comando per disinstallare completamente Astra Trident:

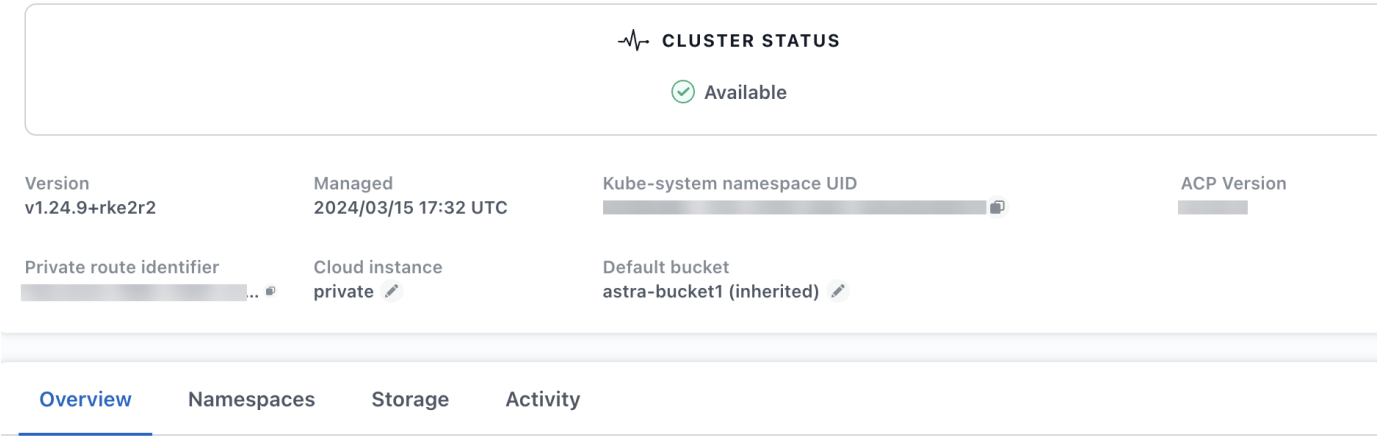
```
./tridentctl uninstall -n trident
```

Non fare "Rimuovere completamente i CRD Astra Trident" Come parte della disinstallazione prima di tentare di attivare nuovamente Astra Control Provivisioner.

Risultato

La funzionalità Astra Control Provisioner è abilitata ed è possibile utilizzare qualsiasi funzionalità disponibile per la versione in esecuzione.

(Solo per gli utenti di Astra Control Center) dopo l'installazione di Astra Control provisioner, il cluster che ospita il provisioner nell'interfaccia utente di Astra Control Center mostrerà un `ACP version` piuttosto che `Trident version` campo e numero della versione installata corrente.



Per ulteriori informazioni

- "Documentazione sugli aggiornamenti di Astra Trident"

Prepara il tuo ambiente per la gestione dei cluster utilizzando Astra Control

Prima di aggiungere un cluster, assicurarsi che siano soddisfatte le seguenti condizioni preliminari. È inoltre necessario eseguire controlli di idoneità per assicurarsi che il cluster sia pronto per essere aggiunto ad Astra Control Center e creare ruoli cluster kubeconfig secondo necessità.

Astra Control consente di aggiungere cluster gestiti da risorse personalizzate (CR) o kubeconfig, a seconda dell'ambiente e delle preferenze.

Prima di iniziare

- **Soddisfare i requisiti ambientali**: Il vostro ambiente soddisfa "requisiti dell'ambiente operativo" Per Astra Control Center.
- **Configura nodi di lavoro**: Assicurarsi che "configurare i nodi di lavoro" nel cluster con i driver di storage appropriati, in modo che i pod possano interagire con lo storage backend.
- **Abilita restrizioni PSA**: Se il cluster ha abilitato l'applicazione di accesso di sicurezza pod, che è standard per i cluster Kubernetes 1,25 e versioni successive, è necessario abilitare le restrizioni PSA nei seguenti spazi dei nomi:
  - `netapp-acc-operator` spazio dei nomi:

```
kubectl label --overwrite ns netapp-acc-operator pod-
security.kubernetes.io/enforce=privileged
```

  - `netapp monitoring` spazio dei nomi:

```
kubectl label --overwrite ns netapp-monitoring pod-
security.kubernetes.io/enforce=privileged
```

- **Credenziali ONTAP**: Per eseguire il backup e il ripristino delle applicazioni con il centro di controllo Astra sono necessarie le credenziali ONTAP e un ID utente e un superutente impostati sul sistema ONTAP di backup.

  Eseguire i seguenti comandi nella riga di comando di ONTAP:

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

- **Requisiti dei cluster gestiti da kubeconfig**: Questi requisiti sono specifici per i cluster di app gestiti da kubeconfig.
  - **Rendere accessibile kubeconfig**: Si ha accesso al "default cluster kubeconfig" quello "la configurazione è stata eseguita durante l'installazione".
  - **Considerazioni sull'autorità di certificazione**: Se si aggiunge il cluster utilizzando un file kubeconfig che fa riferimento a un'autorità di certificazione (CA) privata, aggiungere la seguente riga al `cluster` sezione del file kubeconfig. In questo modo si permette ad Astra Control di aggiungere il cluster:

```
insecure-skip-tls-verify: true
```

  - **Solo Rancher**: Quando si gestiscono i cluster di applicazioni in un ambiente Rancher, modificare il contesto predefinito del cluster di applicazioni nel file kubeconfig fornito da Rancher per utilizzare un contesto del piano di controllo invece del contesto del server API Rancher. In questo modo si riduce il carico sul server API Rancher e si migliorano le performance.
- **Requisiti di Astra Control Provisioner**: Dovresti avere un Astra Control Provisioner configurato correttamente, inclusi i suoi componenti Astra Trident, per gestire i cluster.
  - **Rivedi i requisiti dell'ambiente Astra Trident**: Prima di installare o aggiornare Astra Control provisioner, consulta "frontend, backend e configurazioni host supportati".
  - **Abilitare la funzionalità Astra Control Provisioner**: Si consiglia vivamente di installare Astra Trident 23,10 o versione successiva e di abilitare "Astra Control Provivisioner funzionalità di storage avanzate". Nelle prossime release, Astra Control non supporterà Astra Trident se anche Astra Control Provisioner non è abilitato.
  - **Configurare un backend di archiviazione**: Deve essere presente almeno un backend di archiviazione "Configurato in Astra Trident" sul cluster.

- **Configurare una classe di archiviazione**: Deve essere presente almeno una classe di archiviazione "Configurato in Astra Trident" sul cluster. Se è configurata una classe di archiviazione predefinita, assicurarsi che sia la classe di archiviazione **only** con l'annotazione predefinita.
- **Configurare un controller snapshot volume e installare una classe snapshot volume**: "Installare un controller per lo snapshot del volume" In modo che le snapshot possano essere create in Astra Control. "Creare" almeno uno `VolumeSnapshotClass` Utilizzando Astra Trident.

Eseguire i controlli di idoneità

Eseguire i seguenti controlli di idoneità per assicurarsi che il cluster sia pronto per essere aggiunto ad Astra Control Center.

Fasi

1. Determina la versione di Astra Trident che stai utilizzando:

   ```
   kubectl get tridentversion -n trident
   ```

   Se Astra Trident esiste, l'output è simile a quanto segue:

   ```
   NAME      VERSION
   trident   24.02.0
   ```

   Se Astra Trident non esiste, viene visualizzato un output simile al seguente:

   ```
   error: the server doesn't have a resource type "tridentversions"
   ```

2. Effettuare una delle seguenti operazioni:
   - Se utilizzi Astra Trident 23,01 o versione precedente, utilizza questi elementi "istruzioni" Per effettuare l'aggiornamento a una versione più recente di Astra Trident prima di effettuare l'aggiornamento a Astra Control Provivisioner. È possibile "eseguire un aggiornamento diretto" A Astra Control Provivisioner 24,02 se il tuo Astra Trident si trova all'interno di una finestra a quattro release della versione 24,02. Ad esempio, puoi eseguire l'upgrade direttamente da Astra Trident 23,04 a Astra Control Provisioner 24,02.
   - Se stai eseguendo Astra Trident 23,10 o versione successiva, verifica che Astra Control provisioner sia stato "attivato". Astra Control Provisioner non funzionerà con le versioni di Astra Control Center precedenti alla 23,10. "Aggiorna Astra Control provisioner" In modo che abbia la stessa versione di Astra Control Center che stai effettuando l'aggiornamento per accedere alle funzionalità più recenti.
3. Assicurarsi che tutti i pod (inclusi `trident-acp`) in esecuzione:

   ```
   kubectl get pods -n trident
   ```

4. Determinare se le classi di storage utilizzano i driver Astra Trident supportati. Il nome del provider deve essere `csi.trident.netapp.io`. Vedere il seguente esempio:

```
kubectl get sc
```

Esempio di risposta:

```
NAME                   PROVISIONER            RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
ontap-gold (default)  csi.trident.netapp.io  Delete         Immediate
true                   5d23h
```

Creare un ruolo cluster kubeconfig

Per i cluster gestiti utilizzando kubeconfig, è possibile creare un'autorizzazione limitata o un ruolo di amministratore di autorizzazioni esteso per Astra Control Center. Questa procedura non è necessaria per la configurazione di Astra Control Center, in quanto è già stata configurata una configurazione come parte di "processo di installazione".

Questa procedura consente di creare una configurazione separata se uno dei seguenti scenari si applica al proprio ambiente:

- Si desidera limitare le autorizzazioni di Astra Control sui cluster gestiti
- Si utilizzano più contesti e non è possibile utilizzare il kubeconfig di Astra Control predefinito configurato durante l'installazione oppure un ruolo limitato con un singolo contesto non funziona nell'ambiente

Prima di iniziare

Prima di completare la procedura, assicurarsi di disporre dei seguenti elementi per il cluster che si desidera gestire:

- kubectl v1.23 o versione successiva installata
- Accesso kubectl al cluster che si intende aggiungere e gestire con Astra Control Center

> Per questa procedura, non è necessario l'accesso kubectl al cluster che esegue Astra Control Center.

- Un kubeconfig attivo per il cluster che si intende gestire con i diritti di amministratore del cluster per il contesto attivo

Fasi

1. Creare un account di servizio:
   a. Creare un file di account del servizio denominato `astracontrol-service-account.yaml`.

```
<strong>astracontrol-service-account.yaml</strong>
```

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

b. Applicare l'account del servizio:

```
kubectl apply -f astracontrol-service-account.yaml
```

2. Creare uno dei seguenti ruoli del cluster con autorizzazioni sufficienti per la gestione di un cluster da parte di Astra Control:

Ruolo cluster limitato

Questo ruolo contiene le autorizzazioni minime necessarie per gestire un cluster da Astra Control:

a. Creare un `ClusterRole` file chiamato, ad esempio, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
```

```
    - horizontalpodautoscalers
    - ingresses
    - jobs
    - namespaces
    - networkpolicies
    - persistentvolumeclaims
    - poddisruptionbudgets
    - pods
    - podtemplates
    - replicasets
    - replicationcontrollers
    - replicationcontrollers/scale
    - rolebindings
    - roles
    - secrets
    - serviceaccounts
    - services
    - statefulsets
    - tridentmirrorrelationships
    - tridentsnapshotinfos
    - volumesnapshots
    - volumesnapshotcontents
    verbs:
    - delete

  # Watch resources
  # Necessary to monitor progress
  - apiGroups:
    - ""
    resources:
    - pods
    - replicationcontrollers
    - replicationcontrollers/scale
    verbs:
    - watch

  # Update resources
  - apiGroups:
    - ""
    - build.openshift.io
    - image.openshift.io
    resources:
    - builds/details
    - replicationcontrollers
    - replicationcontrollers/scale
    - imagestreams/layers
```

```
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (Solo per i cluster OpenShift) aggiungere quanto segue alla fine di `astra-admin-account.yaml` file:

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

c. Applicare il ruolo del cluster:

```
kubectl apply -f astra-admin-account.yaml
```

Ruolo cluster esteso

Questo ruolo contiene autorizzazioni estese per un cluster da gestire con Astra Control. È possibile utilizzare questo ruolo se si utilizzano più contesti e non è possibile utilizzare il kubeconfig di Astra Control predefinito configurato durante l'installazione oppure se un ruolo limitato con un singolo contesto non funziona nell'ambiente:

> Quanto segue `ClusterRole` I passaggi sono un esempio generale di Kubernetes. Consultare la documentazione della distribuzione Kubernetes per istruzioni specifiche sull'ambiente in uso.

a. Creare un `ClusterRole` file chiamato, ad esempio, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

b. Applicare il ruolo del cluster:

```
kubectl apply -f astra-admin-account.yaml
```

3. Creare l'associazione del ruolo del cluster all'account del servizio per il ruolo del cluster:

a. Creare un `ClusterRoleBinding` file chiamato `astracontrol-clusterrolebinding.yaml`.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

b. Applicare l'associazione del ruolo del cluster:

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. Creare e applicare il token secret:
   a. Creare un file token secret chiamato `secret-astracontrol-service-account.yaml`.

```
<strong>secret-astracontrol-service-account.yaml</strong>
```

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

   b. Applicare il token secret:

```
kubectl apply -f secret-astracontrol-service-account.yaml
```

5. Aggiungere il token secret all'account del servizio aggiungendo il nome a `secrets` array (l'ultima riga dell'esempio seguente):

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. Elencare i segreti dell'account di servizio, sostituendo `<context>` con il contesto corretto per l'installazione:

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

La fine dell'output dovrebbe essere simile a quanto segue:

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

Gli indici di ciascun elemento in `secrets` l'array inizia con 0. Nell'esempio precedente, l'indice per `astracontrol-service-account-dockercfg-48xhx` sarebbe 0 e l'indice per `secret-astracontrol-service-account` sarebbe 1. Nell'output, annotare il numero dell'indice per il segreto dell'account del servizio. Questo numero di indice è necessario nel passaggio successivo.

7. Generare il kubeconfig come segue:
   a. Creare un `create-kubeconfig.sh` file.
   b. Sostituire `TOKEN_INDEX` all'inizio del seguente script con il valore corretto.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  *-o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
```

```
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

c. Eseguire la sorgente dei comandi per applicarli al cluster Kubernetes.

```
source create-kubeconfig.sh
```

8. (Facoltativo) rinominare il kubeconfig con un nome significativo per il cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

(Anteprima tecnica) Installa Astra Connector per cluster gestiti

I cluster gestiti da Astra Control Center utilizzano Astra Connector per consentire la comunicazione tra il cluster gestito e Astra Control Center. Devi installare Astra Connector su tutti i cluster che desideri gestire.

Installare il connettore Astra

Installi Astra Connector utilizzando i comandi di Kubernetes e i file Custom Resource (CR).

A proposito di questa attività

- Quando esegui questi passaggi, esegui questi comandi sul cluster che desideri gestire con Astra Control.
- Se si utilizza un host Bastion, eseguire questi comandi dalla riga di comando dell'host Bastion.

Prima di iniziare

- Devi accedere al cluster da gestire con Astra Control.
- Sono necessarie autorizzazioni di amministratore di Kubernetes per installare l'operatore Astra Connector sul cluster.

> Se il cluster è configurato con l'imposizione dell'ammissione di sicurezza pod, che è l'impostazione predefinita per i cluster Kubernetes 1,25 e versioni successive, è necessario abilitare le restrizioni PSA sugli spazi dei nomi appropriati. Fare riferimento a. "Prepara il tuo ambiente per la gestione dei cluster utilizzando Astra Control" per istruzioni.

Fasi

1. Installa l'operatore Astra Connector sul cluster che desideri gestire con Astra Control. Quando si esegue questo comando, lo spazio dei nomi `astra-connector-operator` viene creato e la configurazione viene applicata allo spazio dei nomi:

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. Verificare che l'operatore sia installato e pronto:

```
kubectl get all -n astra-connector-operator
```

3. Ottieni un token API da Astra Control. Fare riferimento a. "Documentazione di Astra Automation" per istruzioni.
4. Creare un segreto utilizzando il token. Sostituisci <API_TOKEN> con il token ricevuto da Astra Control:

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. Crea un Docker Secret da usare per estrarre l'immagine di Astra Connector. Sostituire i valori tra parentesi <> con le informazioni dell'ambiente:

> Puoi trovare il <ASTRA_CONTROL_ACCOUNT_ID> nell'interfaccia utente web di Astra Control. Nell'interfaccia utente Web, selezionare l'icona della figura in alto a destra nella pagina e selezionare **accesso API**.

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. Creare il file Astra Connector CR e assegnargli un nome `astra-connector-cr.yaml`. Aggiorna i valori tra parentesi <> per farli corrispondere all'ambiente Astra Control e alla configurazione del cluster:
   - <ASTRA_CONTROL_ACCOUNT_ID>: Ottenuto dall'interfaccia utente web Astra Control durante la fase precedente.

- <CLUSTER_NAME>: Il nome che il cluster deve essere assegnato in Astra Control.
- <ASTRA_CONTROL_URL>: L'URL dell'interfaccia utente web di Astra Control. Ad esempio:

```
https://astra.control.url
```

```yaml
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. Dopo aver popolato il `astra-connector-cr.yaml` File con i valori corretti, applicare il CR:

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. Verificare che il connettore Astra sia completamente distribuito:

```
kubectl get all -n astra-connector
```

9. Verifica che il cluster sia registrato con Astra Control:

```
kubectl get astraconnectors.astra.netapp.io -A
```

L'output dovrebbe essere simile a quanto segue:

```
NAMESPACE          NAME               REGISTERED   ASTRACONNECTORID
STATUS
astra-connector    astra-connector    true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. Verificare che il cluster compaia nell'elenco dei cluster gestiti nella pagina **cluster** dell'interfaccia utente Web Astra Control.

Aggiungere un cluster

Per iniziare a gestire le tue applicazioni, Aggiungi un cluster Kubernetes e gestilo come risorsa di calcolo. Devi aggiungere un cluster per Astra Control Center per scoprire le tue applicazioni Kubernetes.

> Si consiglia ad Astra Control Center di gestire il cluster su cui viene implementato prima di aggiungere altri cluster ad Astra Control Center da gestire. La gestione del cluster iniziale è necessaria per inviare i dati Kublemetrics e i dati associati al cluster per metriche e troubleshooting.

Prima di iniziare

- Prima di aggiungere un cluster, esaminare ed eseguire le operazioni necessarie "attività prerequisite".
- Se stai utilizzando un driver SAN ONTAP, assicurati che multipath sia abilitato su tutti i tuoi cluster Kubernetes.

Fasi

1. Spostarsi dal menu Dashboard o Clusters:
   - Da **Dashboard** in Resource Summary (Riepilogo risorse), selezionare **Add** (Aggiungi) dal pannello Clusters (Clusters).
   - Nell'area di navigazione a sinistra, selezionare **Clusters**, quindi selezionare **Add Cluster** (Aggiungi cluster) dalla pagina Clusters (Cluster).
2. Nella finestra **Add Cluster** che si apre, caricare un `kubeconfig.yaml` archiviare o incollare il contenuto di a. `kubeconfig.yaml` file.

   > Il `kubeconfig.yaml` il file deve includere **solo le credenziali del cluster per un cluster**.

   > Se crei il tuo `kubeconfig` file, è necessario definire solo **un** elemento di contesto al suo interno. Fare riferimento a. "Documentazione Kubernetes" per informazioni sulla creazione `kubeconfig` file. Se hai creato un kubeconfig per un ruolo cluster limitato utilizzando "questo processo", assicurarsi di caricare o incollare il kubeconfig in questa fase.

3. Fornire un nome di credenziale. Per impostazione predefinita, il nome della credenziale viene compilato automaticamente come nome del cluster.
4. Selezionare **Avanti**.
5. Selezionare la classe di storage predefinita da utilizzare per il cluster Kubernetes e selezionare **Avanti**.

> Scegli una classe di storage configurata in Astra Control Provivisioner e supportata dallo storage ONTAP.

6. Esaminare le informazioni e, se tutto sembra buono, selezionare **Aggiungi**.

Risultato

Il cluster passa allo stato **Discovering** e quindi passa a **Healthy**. Ora stai gestendo il cluster con Astra Control Center.

> Dopo aver aggiunto un cluster da gestire in Astra Control Center, l'implementazione dell'operatore di monitoraggio potrebbe richiedere alcuni minuti. Fino a quel momento, l'icona di notifica diventa rossa e registra un evento **Monitoring Agent Status Check Failed** (controllo stato agente non riuscito). È possibile ignorarlo, perché il problema si risolve quando Astra Control Center ottiene lo stato corretto. Se il problema non si risolve in pochi minuti, accedere al cluster ed eseguire `oc get pods -n netapp-monitoring` come punto di partenza. Per eseguire il debug del problema, è necessario esaminare i registri dell'operatore di monitoraggio.

Abilitare l'autenticazione su un backend di storage ONTAP

Il centro di controllo Astra offre due modalità di autenticazione di un backend ONTAP:

- **Autenticazione basata su credenziali**: Nome utente e password di un utente ONTAP con le autorizzazioni richieste. Per garantire la massima compatibilità con le versioni di ONTAP, è necessario utilizzare un ruolo di accesso di sicurezza predefinito, ad esempio admin o vsadmin.
- **Autenticazione basata su certificato**: Il centro di controllo Astra può anche comunicare con un cluster ONTAP utilizzando un certificato installato sul back-end. Utilizzare il certificato client, la chiave e il certificato CA attendibile, se utilizzato (consigliato).

È possibile aggiornare in seguito i back-end esistenti per passare da un tipo di autenticazione a un altro metodo. È supportato un solo metodo di autenticazione alla volta.

Abilitare l'autenticazione basata su credenziali

Astra Control Center richiede le credenziali per un cluster con ambito `admin` Per comunicare con il backend ONTAP. È necessario utilizzare ruoli standard predefiniti, ad esempio `admin`. Ciò garantisce la compatibilità con le future release di ONTAP che potrebbero esporre le API delle funzionalità da utilizzare nelle future release di Astra Control Center.

> Un ruolo di accesso di sicurezza personalizzato può essere creato e utilizzato con Astra Control Center, ma non è consigliato.

Un esempio di definizione di backend è simile al seguente:

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

La definizione di backend è l'unica posizione in cui le credenziali vengono memorizzate in testo normale. La creazione o l'aggiornamento di un backend è l'unico passaggio che richiede la conoscenza delle credenziali. Pertanto, si tratta di un'operazione di sola amministrazione, che deve essere eseguita da Kubernetes o dall'amministratore dello storage.

Abilitare l'autenticazione basata su certificato

Il centro di controllo Astra può utilizzare i certificati per comunicare con i backend ONTAP nuovi ed esistenti. Inserire le seguenti informazioni nella definizione di backend.

- `clientCertificate`: Certificato del client.
- `clientPrivateKey`: Chiave privata associata.
- `trustedCACertificate`: Certificato CA attendibile. Se si utilizza una CA attendibile, è necessario fornire questo parametro. Questa operazione può essere ignorata se non viene utilizzata alcuna CA attendibile.

È possibile utilizzare uno dei seguenti tipi di certificati:

- Certificato autofirmato
- Certificato di terze parti

Abilitare l'autenticazione con un certificato autofirmato

Un workflow tipico prevede i seguenti passaggi.

Fasi

1. Generare un certificato e una chiave del client. Durante la generazione, impostare il nome comune (CN) sull'utente ONTAP per l'autenticazione come.

   ```
   openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
   -out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
   ```

2. Installare il certificato client di tipo `client-ca` E sul cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. Verificare che il ruolo di accesso di sicurezza di ONTAP supporti il metodo di autenticazione del certificato.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

4. Verificare l'autenticazione utilizzando il certificato generato. Sostituire <LIF di gestione ONTAP> e <vserver name> con l'IP LIF di gestione e il nome SVM. Assicurarsi che la politica di servizio di LIF sia impostata su `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-
name>"><vserver-get></vserver-get></netapp>
```

5. Utilizzando i valori ottenuti dal passaggio precedente, aggiungere il backend di storage nell'interfaccia utente di Astra Control Center.

Abilitare l'autenticazione con un certificato di terze parti

Se si dispone di un certificato di terze parti, è possibile configurare l'autenticazione basata su certificato con questa procedura.

Fasi

1. Generare la chiave privata e la CSR:

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem
-out ontap_cert_request.csr -keyout ontap_cert_request.key -addext
"subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>"
```

2. Passare la CSR alla CA di Windows (CA di terze parti) e rilasciare il certificato firmato.
3. Scarica il certificato firmato e chiamalo `ontap_signed_cert.crt'
4. Esportare il certificato root dalla CA di Windows (CA di terze parti).
5. Assegnare un nome al file `ca_root.crt`

   A questo punto, sono disponibili i seguenti tre file:

- **Chiave privata**: `ontap_signed_request.key` (Chiave corrispondente al certificato del server in ONTAP). È necessario durante l'installazione del certificato del server).
- **Certificato firmato**: `ontap_signed_cert.crt` (Questo è anche chiamato *certificato del server* in ONTAP).
- **Certificato CA root**: `ca_root.crt` (Questo è anche chiamato *certificato server-ca* in ONTAP).

6. Installare questi certificati in ONTAP. Generare e installare `server` e. `server-ca` Certificati su ONTAP.

Espandere per sample.yaml

```
# Copy the contents of ca_root.crt and use it here.

security certificate install -type server-ca

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for
future reference.

The installed certificate's CA and serial number for reference:

CA:
serial:

The certificate's generated name for reference:


===

# Copy the contents of ontap_signed_cert.crt and use it here. For
key, use the contents of ontap_cert_request.key file.
security certificate install -type server
Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----

Please enter Private Key: Press <Enter> when done

-----BEGIN PRIVATE KEY-----
<private key details>
-----END PRIVATE KEY-----

Enter certificates of certification authorities (CA) which form the
certificate chain of the server certificate. This starts with the
issuing CA certificate of the server certificate and can range up to
the root CA certificate.
Do you want to continue entering root and/or intermediate
```

```
certificates {y|n}: n

The provided certificate does not have a common name in the subject
field.
Enter a valid common name to continue installation of the
certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital
certificate for future reference.
The installed certificate's CA and serial number for reference:
CA:
serial:
The certificate's generated name for reference:


==
# Modify the vserver settings to enable SSL for the installed
certificate

ssl modify -vserver <vserver_name> -ca <CA>  -server-enabled true
-serial <serial number>       (security ssl modify)

==
# Verify if the certificate works fine:

openssl s_client -CAfile ca_root.crt -showcerts -servername server
-connect <ONTAP_CLUSTER_FQDN_NAME>:443
CONNECTED(00000005)
depth=1 DC = local, DC = umca, CN = <CA>
verify return:1
depth=0
verify return:1
write W BLOCK
---
Certificate chain
0 s:
   i:/DC=local/DC=umca/<CA>

-----BEGIN CERTIFICATE-----
<Certificate details>
```

7. Creare il certificato client per lo stesso host per le comunicazioni senza password. Il centro di controllo Astra utilizza questo processo per comunicare con ONTAP.
8. Generare e installare i certificati client su ONTAP:

Espandere per sample.yaml

```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"

Copy the content of ontap_test_client.pem file and use it in the
below command:
security certificate install -type client-ca -vserver <vserver_name>

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<Certificate details>
-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for
future reference.
The installed certificate's CA and serial number for reference:

CA:
serial:
The certificate's generated name for reference:


==

ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)

# Setting permissions for certificates
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>

security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>

==

#Verify passwordless communication works fine with the use of only
certificates:

curl --cacert ontap_signed_cert.crt  --key ontap_test_client.key
--cert ontap_test_client.pem
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
{
```

```
"records": [
{
"uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
"name": "<aggr_name>",
"node": {
"uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
"name": "<node_name>",
"_links": {
"self": {
"href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
}
}
},
"_links": {
"self": {
"href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-
4bf5378b41bd"
}
}
}
],
"num_records": 1,
"_links": {
"self": {
"href": "/api/storage/aggregates"
}
}
}%
```


9. Aggiungere il backend dello storage nell'interfaccia utente di Astra Control Center e fornire i seguenti valori:
   - **Certificato client**: ontap_test_client.pem
   - **Chiave privata**: ontap_test_client.key
   - **Certificato CA attendibile**: ontap_signed_cert.crt

Aggiungere un backend di storage

Dopo aver impostato le credenziali o le informazioni di autenticazione del certificato, è possibile aggiungere un backend di storage ONTAP esistente a Astra Control Center per gestire le risorse.

La gestione dei cluster di storage in Astra Control come back-end dello storage consente di ottenere collegamenti tra volumi persistenti (PVS) e il back-end dello storage, oltre a metriche di storage aggiuntive.

L'aggiunta e la gestione dei backend di storage ONTAP in Astra Control Center sono opzionali quando si utilizza la tecnologia NetApp SnapMirror, se hai abilitato Astra Control Provivisioner.

Fasi

1. Dal pannello di controllo nell'area di navigazione a sinistra, selezionare **Backend**.
2. Selezionare **Aggiungi**.
3. Nella sezione Use existing della pagina Add storage backend, selezionare **ONTAP**.
4. Selezionare una delle seguenti opzioni:
   - **Usa credenziali amministratore**: Inserire l'indirizzo IP di gestione del cluster ONTAP e le credenziali di amministratore. Le credenziali devono essere credenziali a livello di cluster.

     > L'utente di cui si inseriscono le credenziali deve disporre di `ontapi` Metodo di accesso all'accesso dell'utente abilitato in Gestione di sistema di ONTAP sul cluster ONTAP. Se si intende utilizzare la replica SnapMirror, applicare le credenziali utente con il ruolo "admin", che dispone dei metodi di accesso `ontapi` e. `http`, Sui cluster ONTAP di origine e di destinazione. Fare riferimento a. "Gestire gli account utente nella documentazione di ONTAP" per ulteriori informazioni.

   - **Usa un certificato**: Carica il certificato `.pem` file, la chiave del certificato `.key` e, facoltativamente, il file dell'autorità di certificazione.
5. Selezionare **Avanti**.
6. Confermare i dettagli del back-end e selezionare **Manage** (Gestisci).

Risultato

Il backend viene visualizzato in `online` indicare nell'elenco le informazioni di riepilogo.

> Potrebbe essere necessario aggiornare la pagina per visualizzare il backend.

Aggiungi un bucket

È possibile aggiungere un bucket utilizzando l'interfaccia utente di Astra Control o. "API di controllo Astra". L'aggiunta di provider di bucket di archivi di oggetti è essenziale se si desidera eseguire il backup delle applicazioni e dello storage persistente o se si desidera clonare le applicazioni tra cluster. Astra Control memorizza i backup o i cloni nei bucket dell'archivio di oggetti definiti dall'utente.

Non è necessario un bucket in Astra Control se si esegue il cloning della configurazione dell'applicazione e dello storage persistente sullo stesso cluster. La funzionalità di snapshot delle applicazioni non richiede un bucket.

Prima di iniziare

- Assicurati di avere un bucket raggiungibile dai cluster gestiti da Astra Control Center.
- Assicurarsi di disporre delle credenziali per il bucket.
- Assicurarsi che la benna sia di uno dei seguenti tipi:
  - NetApp ONTAP S3
  - NetApp StorageGRID S3
  - Microsoft Azure
  - Generico S3

Amazon Web Services (AWS) e Google Cloud Platform (GCP) utilizzano il tipo di bucket S3 generico.

Sebbene Astra Control Center supporti Amazon S3 come provider di bucket S3 generico, Astra Control Center potrebbe non supportare tutti i vendor di archivi di oggetti che rivendicano il supporto S3 di Amazon.

Fasi

1. Nell'area di navigazione a sinistra, selezionare **Bucket**.
2. Selezionare **Aggiungi**.
3. Selezionare il tipo di bucket.

   Quando si aggiunge un bucket, selezionare il bucket provider corretto e fornire le credenziali corrette per tale provider. Ad esempio, l'interfaccia utente accetta come tipo NetApp ONTAP S3 e accetta le credenziali StorageGRID; tuttavia, questo causerà l'errore di tutti i backup e ripristini futuri dell'applicazione che utilizzano questo bucket.

4. Inserire un nome bucket esistente e una descrizione opzionale.

   Il nome e la descrizione del bucket vengono visualizzati come una posizione di backup che è possibile scegliere in seguito quando si crea un backup. Il nome viene visualizzato anche durante la configurazione del criterio di protezione.

5. Inserire il nome o l'indirizzo IP dell'endpoint S3.
6. In **Seleziona credenziali**, selezionare la scheda **Aggiungi** o **Usa esistente**.
   - Se si sceglie **Aggiungi**:
     i. Immettere un nome per la credenziale che la distingue dalle altre credenziali in Astra Control.
     ii. Inserire l'ID di accesso e la chiave segreta incollando il contenuto dagli Appunti.
   - Se si sceglie **Usa esistente**:
     i. Selezionare le credenziali esistenti che si desidera utilizzare con il bucket.
7. Selezionare `Add`.

   Quando si aggiunge un bucket, Astra Control contrassegna un bucket con l'indicatore bucket predefinito. Il primo bucket creato diventa quello predefinito. Con l'aggiunta di bucket, è possibile decidere in un secondo momento "impostare un altro bucket predefinito".